REPORT
ON
Professional Internship (PI)
WITH
Polytechnique Montréal, Canada

Submitted by: DENG ZICHAO
Matriculation Number: U1420707G

Supervisor: Mr. Yann-Gaël Guéhéneuc
NTU Tutor: Prof. Mo Yilin

School of Electrical & Electronic Engineering

Table of Contents

CHAPTER **PAGE**

Abstract

Ptidej is the short of Pattern Trace Identification, Detection, and Enhancement in Java. The Ptidej team aims at developing theories, methods, and tools to understand, evaluate, and improve the quality of software systems by promoting the use of idioms, design patterns, and architectural patterns.

My work is mainly to improve the quality of the online version of Ptidej, which is “brunch”. I focus on some web development programming and some database programming as well.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor, Mr. Yann-Gaël Guéhéneuc for giving me such a great research opportunity in the department of computer engineering and software engineering of Polytechnique Montreal.

Along my internship journey, I sincerely thank him for his patience, valuable guidance and instructions, which make me gain a lot knowledge and hands on experience regarding the design patterns, web application development, database programming, etc.

What's more, I want to specially express my thanks to Ms. Nathalie PELLETIER, who contacts my host university, Nanyang Technological University, to provide this internship for me and makes all the things happen. I am grateful for her warm and detailed care for me during my stay in Canada.

Last but not least, I appreciate that my host university, Nanyang Technological University, give me a chance to do this Professional Internship (PI) and learn knowledge and experience which can be applied to the real computer/software engineering industrial field and benefit to my future career.

Thanks to my NTU tutor, Prof. Mo Yilin for supervising me and keeping update to my internship progress, which help me a lot on completing my internship successfully.

Chapter 1. Introduction to Basic Information

1.1 Introduction to Polytechnique Montreal

The University of Montreal was established in 1878 and comprised two separate school, the Polytechnique School and HEC Montreal. It has become the largest university of Quebec and the second largest university in Canada. There are more than 60,000 students from all over the world in the University of Montreal and there are more than 11,000 degrees awarded each year.

Research study is the crucial academic part of the university, which makes the university profit more than a half-billion dollars. In terms of this, the University of Montreal becomes the third most dynamic university in Canada depend on the research study and has been the most dynamic university in Quebec from year 2005 till now.

Especially, the Polytechnique Montreal was established in 1873 and has become one of the best education institutions in the filed of engineering education and research. What's more, the Polytechnique Montreal is the top university in Quebec in term of the student quantity as well as the content of research programs.

The Polytechnique Montreal offers various kinds of courses, trainings and research opportunities in the area of Aerospace Engineering, Biomedical Engineering, Chemical Engineering, Civil, Geological and/or Mining Engineering, Computer and Software Engineering, Electrical Engineering, Mechanical Engineering, Physics Engineering and/or Materials Science and etc. In the fields mentioned above, the research activities of the Polytechnique Montreal take up almost one-quarter of university research programs in Quebec. Last but not least, the Polytechnique Montreal also touches a few most intensive and confidential research across about 60 research areas and a bunch of famous and dedicated researchers.

The emblem of the Polytechnique of Montreal is a bee surrounding by a circle gear crossed by the steel beam. In explanation, the bee in the circle stands for the engineers who are organized well and work together in a detailed and planned manner. The steel possible represents the fundamental engineering academy: civil engineering. In addition, the gear outside the bee stands for the engineering industry flourish at the end of 19th century, during that period, engineers played a crucial role in the industry development. At last, the laurel wreath represents excellence. This emblem accompanies with a slogan "UT TENSIO SIC VIS". This slogan sources from a law of material resistance called "Hooke's law", in detailed, "Elongation is proportional to force". We can apply this law generally and get that:" The outcome is proportional to efforts made". This slogan is trying to encourage each student in Polytechnique of Montreal to make progress in research.

The Figure 1 shows the picture of Polytechnique of Montreal and Figure 2 shows the emblem of Polytechnique of Montreal.

Figure 1. The Polytechnique of Montreal

Figure 2. Emblem of the Polytechnique of Montreal

1.2 Introduction to the Department

There are about 10 labs in the department of computer engineering and software engineering. And the experts and students work on various fields including computer architecture and semiconductors, industrial computing and numerical control, networking and telematics, software engineering, artificial intelligence and etc. What's more, students trained here has excellent academic technology knowledge as well as brilliant analyze and problem-solving ability. A lot of research projects here lead to tackle the real-world issues or problems.

1.2.1 Introduction to PolyMORSE

The PolyMORSE stands for POLYtechnique MOntreal Researchers in Software Engineering. The domain research interest of PolyMORSE is to boost the understanding of collaborative and cognitive programs in software development and investigate the integration of user-friendly approaches in the software engineering and software development life cycle. It is the mission and goal for PolyMORSE to improve and develop more effective engineering process products which can be integrated into different collaborative and multidisciplinary conditions.

There are a few laboratories under the PolyMORSE:

- Soccer Lab: SOftware Cost-effective Change and Evolution Research lab
- SWAT: SoftWare Analytics and Technologies Lab
- MCIS: Maintenance, Construction and Intelligence of Software
- Ptidej Team: Pattern Trace Identification, Detection, and Enhancement in Java

I am involved into the Ptidej team and do my research internship for 4 months.

1.2.2 Introduction to Ptidej Team

Ptidej is the short of Pattern Trace Identification, Detection, and Enhancement in Java. The Ptidej team aims at developing theories, methods, and tools to understand, evaluate, and improve the quality of software systems by promoting the use of idioms, design patterns, and architectural patterns. The Ptidej team would like to formalize patterns, to identify occurrences of patterns, and to improve the identified occurrences. What's more, the Ptidej team also aims to evaluate experimentally the impact of patterns on the quality of software systems. The Ptidej team has developed various tools, especially the Ptidej tool suite and Taupe, to evaluate and to enhance the quality of software systems, by promoting the use of patterns, either at the language-, design-, or architectural- levels. [1]

The Figure 3 shows the logo of the Ptidej team.

Ptidej
Pattern Trace Identification, Detection, and Enhancement in Java

Figure 3. The logo of the Ptidej team

Chapter 2. Literature Review of Ptidej Tool Suite

2.1 General Review

In general, the Ptidej is developed in Java language. In total, there are several modules, more than 300 packages, 2000 classes and 300 interfaces involved in the Ptidej tool suite. The Ptidej tool suite actually is developed to evaluate and to enhance the quality of software systems, especially the object-oriented programs by promoting the usage of idioms, design patterns and architectural patterns.

The ptidej tool suite offers an architecture to model the software systems/object-oriented programs and then, analyze the programs by applying a few algorithms including the data conversion algorithms, third-party programs and etc. The Ptidej tool suite is popular around the world for analyzing and improving the quality of software systems.

2.2 Architecture of the Ptidej tool suite

The Ptidej tool suite is a quite complex project. Here I just focus on two main parts of the Ptidej tool suite: The Meta-model and the User interface.

2.2.1 The Meta-Model Architecture

The Meta-model is the PADL, which is the short of Pattern and Abstractlevel Description Language. The PADL is used to describe the structure of motifs, which is the "Solution" parts in the definition of design pattern in the process of analyzing software systems, especially the object-oriented programs.

There is a library of design motifs, the "solution" part in the definition of design patterns, such as Singleton Pattern, Prototype Pattern, Strategy Pattern, Visitor Pattern, MVC Pattern and etc.

In order to configure the models of software systems from various representation of source code such as the AOL, Java class files as well as C++ source codes, a few parsers are used.

What’s more, there is a library of software metrics called POM (Primitives, Operators, Metrics), which is used to apply famous metrics after modeling of the software systems including Chidamber and Kemerer’s metrics and etc.

In addition, there is also a library of generators and analyses applying on the model of programs as well as the design motifs.

PTIDEJ SOLVER, which is an explanation-based constraint solver, is used to detect some micro-architectures in the model of programs, which are quite similar to design motifs. What’s more, it returns different roles the classes play in the models of programs.

Last but not least, CAFFEINE, which is a dynamic analyzer for Java, is used to make the definitions of relationships among various classes accurately on the basis of a Prolog engine as well as the Java debug interface.

When the user posts a query which is a single goal generally, the Prolog program executes and the Prolog engine is used to find a resolution refutation of the negated query. [2]

The Figure 4 show the structure of the Meta-model architecture.

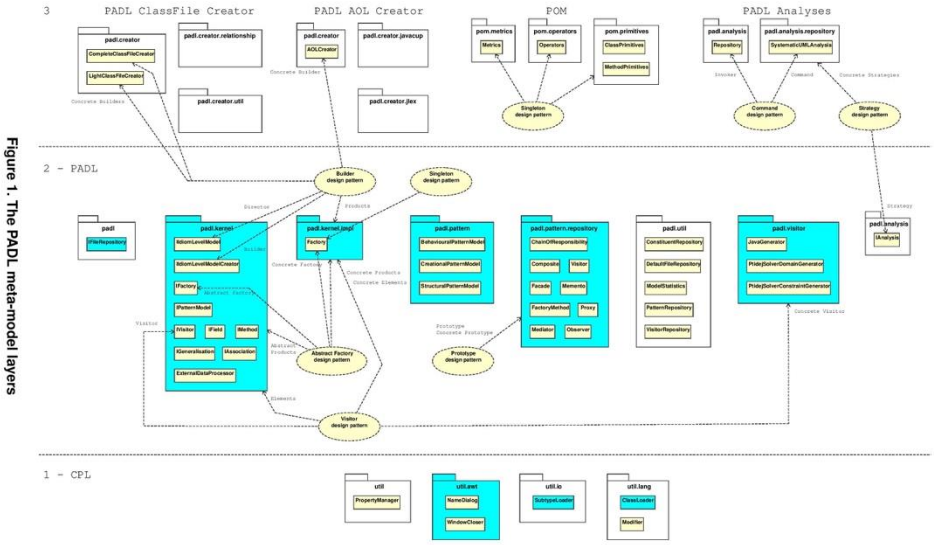


Figure 4. The PADL Meta-model layers

2.2.2 The User Interface Architecture

There is a library of graphic widgets called PTIDEJ UI, which is used to show these dynamic data generated from CAFFEINE as well as the design motifs of the software systems.

What’s more, there are other user-interfaces to display the services offered by the Ptidej tool suite:

- Analyze and create the models of software systems or object-oriented programs
- Display and visualize the created models of programs
- From executing the programs, the models of programs are enhanced with the generation of dynamic data
- Find the micro-architectures by the PTIDEJ SOLVER and display these micro-architectures, which are similar to the design motifs in the software systems.
- Execute some other tools inside the Ptidej tool suite including some generators and analyses.

The Figure 5 show the structure of user-interface architecture.

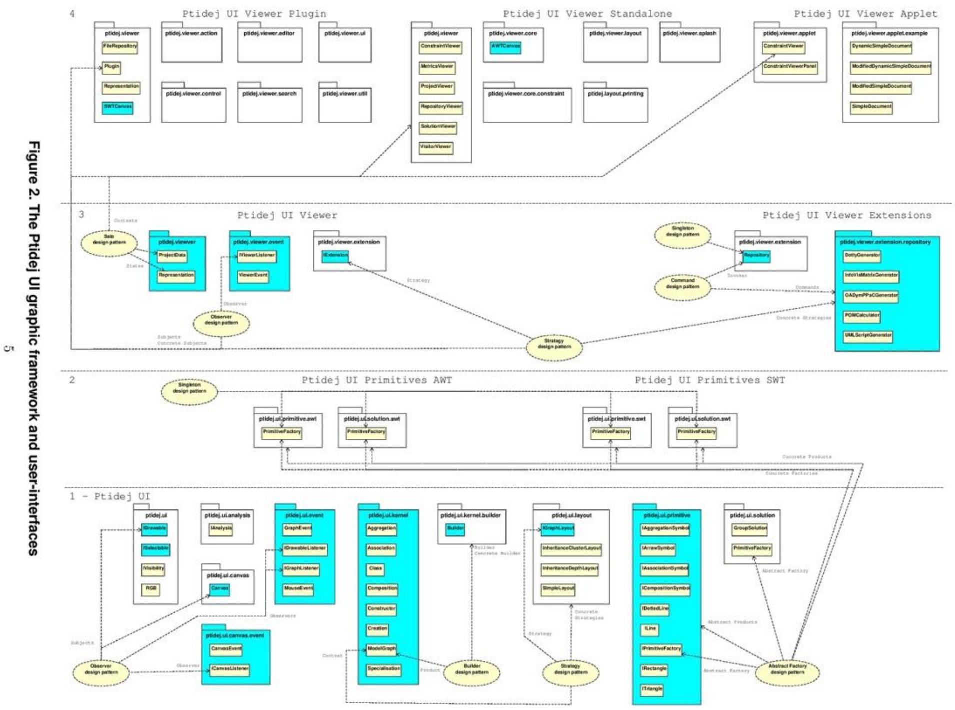


Figure 5. The Ptidej UI graphic framework and user-interfaces

2.3 Highlight a few services of Ptidej

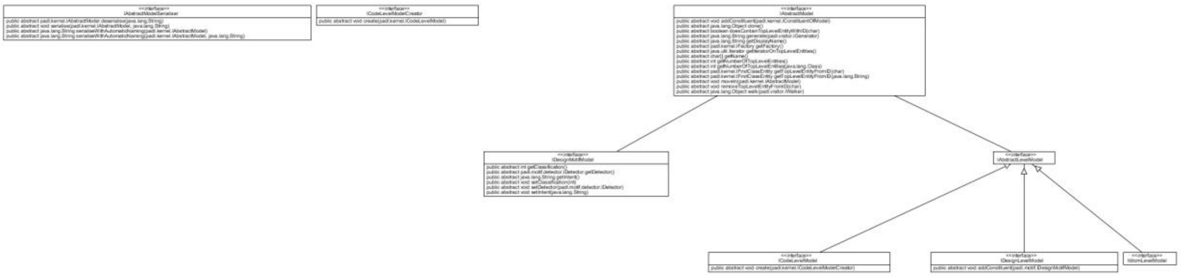
Although the Ptidej tool suite contains a lot of models, I want to introduce a few crucial services offered by the Ptidej tool suite.

2.3.1 PADL

PADL is the short of Pattern and Abstract-level Description Language, which is used to identify and describe the design motifs, the “solution” part in the design patterns of software systems. What’s more, PADL describes the object-oriented programs at various level of abstraction. There are totally 4 levels of abstraction:

- ICodeLevelModel: This is the “raw” model of object-oriented programs, which only contains the data directly extracted from the source codes including the Java source code, C/C++ source code and Java byte code.
- IIdiomLevelModel: In the models of programs, some idioms have been reified including the binary-class relationship.
- IDesignLevelModel: the design information such as the occurrences of design patterns is visible and understand by people at this level of abstraction.
- IDesignMotif: the design motifs, the “solution” in the definition of design patterns, are represented in the models of programs.

The Figure 6 shows the PADL classes hierarchy.



<<interface>> IAbstractModelSerialiser
<<interface>> ICodeLevelModelCreator
<<interface>> IAbstractModel
<<interface>> IDesignMotifModel
<<interface>> IAbstractLevelModel
<<interface>> ICodeLevelModel
<<interface>> IDesignLevelModel
<<interface>> IIdiomLevelModel

Figure 6. PADL classes hierarchy

2.3.2 Ptidej Solver

The service, Ptidej Solver, is used to identify the design patterns inside the models of programs and different roles which classes play in the design pattern by using an explanation-based constraint. About 20 design patterns can be identified:

- Composite Design Pattern: The composite design pattern is used when we consider a group of objects as a single object.
- Chain of Responsibility Design pattern: It consists a command object and a list of processing objects.
- Factory Design Pattern: This pattern is used when we would like to instantiate a object but not specify the class of this object.

The Figure 7 shows the Ptidej Solver classes hierarchy.

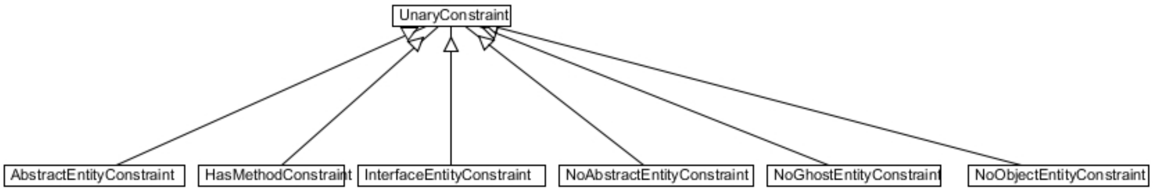


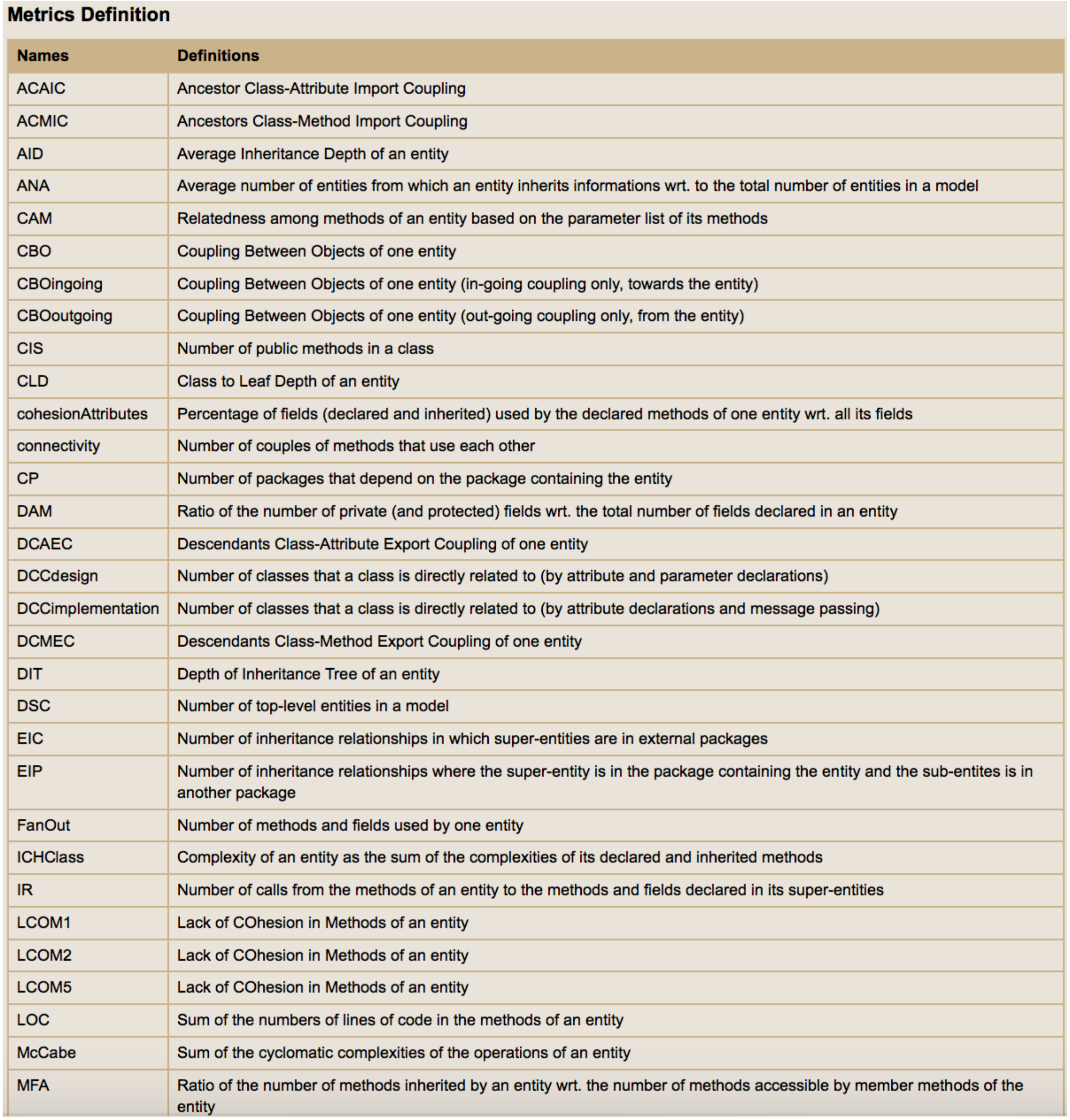
Figure 7. Ptidej Solver classes hierarchy

2.3.3 POM

POM is the short of Primitives, Operators, Metrics. This service is to apply various metrics on the models of programs. Generally, there are two types of metrics including unary and binary. The unary metric is our focus. There are lots of metrics:

- CAM: Relatedness among methods of an entity based on the parameter list of its methods
- DIT: Depth of Inheritance Tree of an entity
- LOC: Sum of the numbers of lines of code in the methods of an entity

The Figure 8 shows some metrics definition. All the resources are available in http://wiki.ptidej.net/doku.php?id=pom.



Metrics Definition

Names	Definitions
ACAIC	Ancestor Class-Attribute Import Coupling
ACMIC	Ancestors Class-Method Import Coupling
AID	Average Inheritance Depth of an entity
ANA	Average number of entities from which an entity inherits informations wrt. to the total number of entities in a model
CAM	Relatedness among methods of an entity based on the parameter list of its methods
CBO	Coupling Between Objects of one entity
CBOingoing	Coupling Between Objects of one entity (in-going coupling only, towards the entity)
CBOoutgoing	Coupling Between Objects of one entity (out-going coupling only, from the entity)
CIS	Number of public methods in a class
CLD	Class to Leaf Depth of an entity
cohesionAttributes	Percentage of fields (declared and inherited) used by the declared methods of one entity wrt. all its fields
connectivity	Number of couples of methods that use each other
CP	Number of packages that depend on the package containing the entity
DAM	Ratio of the number of private (and protected) fields wrt. the total number of fields declared in an entity
DCAEC	Descendants Class-Attribute Export Coupling of one entity
DCCdesign	Number of classes that a class is directly related to (by attribute and parameter declarations)
DCCimplementation	Number of classes that a class is directly related to (by attribute declarations and message passing)
DCMEC	Descendants Class-Method Export Coupling of one entity
DIT	Depth of Inheritance Tree of an entity
DSC	Number of top-level entities in a model
EIC	Number of inheritance relationships in which super-entities are in external packages
EIP	Number of inheritance relationships where the super-entity is in the package containing the entity and the sub-entites is in another package
FanOut	Number of methods and fields used by one entity
ICHClass	Complexity of an entity as the sum of the complexities of its declared and inherited methods
IR	Number of calls from the methods of an entity to the methods and fields declared in its super-entities
LCOM1	Lack of COhesion in Methods of an entity
LCOM2	Lack of COhesion in Methods of an entity
LCOM5	Lack of COhesion in Methods of an entity
LOC	Sum of the numbers of lines of code in the methods of an entity
McCabe	Sum of the cyclomatic complexities of the operations of an entity
MFA	Ratio of the number of methods inherited by an entity wrt. the number of methods accessible by member methods of the entity

Figure 8. POM metrics definitions

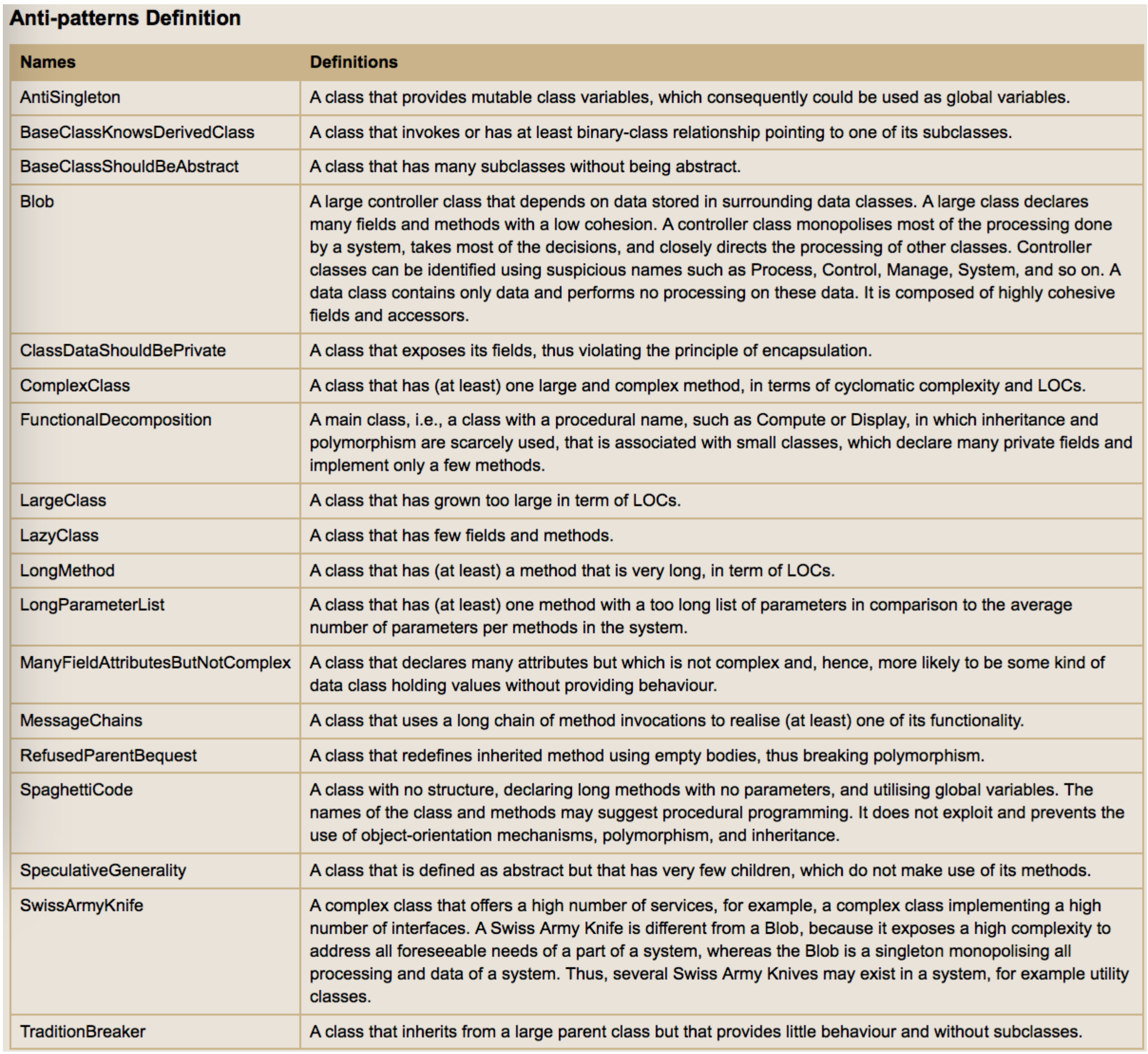
2.3.4 SAD

SAD is the short of Software Architectural Defects. It depends on the outcome of POM metrics applied on the models of programs. SAD is used to identify a few code smells and anti-patterns:

- AntiSingleton: A class that provides mutable class variables, which consequently could be used as global variables.

- LargeClass: A class that has grown too large in terms of LOCs.
- LazyClass: A class that has few fields and methods. [3]

The Figure 9 shows a few anti-patterns definitions.



Anti-patterns Definition

Names	Definitions
AntiSingleton	A class that provides mutable class variables, which consequently could be used as global variables.
BaseClassKnowsDerivedClass	A class that invokes or has at least binary-class relationship pointing to one of its subclasses.
BaseClassShouldBeAbstract	A class that has many subclasses without being abstract.
Blob	A large controller class that depends on data stored in surrounding data classes. A large class declares many fields and methods with a low cohesion. A controller class monopolises most of the processing done by a system, takes most of the decisions, and closely directs the processing of other classes. Controller classes can be identified using suspicious names such as Process, Control, Manage, System, and so on. A data class contains only data and performs no processing on these data. It is composed of highly cohesive fields and accessors.
ClassDataShouldBePrivate	A class that exposes its fields, thus violating the principle of encapsulation.
ComplexClass	A class that has (at least) one large and complex method, in terms of cyclomatic complexity and LOCs.
FunctionalDecomposition	A main class, i.e., a class with a procedural name, such as Compute or Display, in which inheritance and polymorphism are scarcely used, that is associated with small classes, which declare many private fields and implement only a few methods.
LargeClass	A class that has grown too large in term of LOCs.
LazyClass	A class that has few fields and methods.
LongMethod	A class that has (at least) a method that is very long, in term of LOCs.
LongParameterList	A class that has (at least) one method with a too long list of parameters in comparison to the average number of parameters per methods in the system.
ManyFieldAttributesButNotComplex	A class that declares many attributes but which is not complex and, hence, more likely to be some kind of data class holding values without providing behaviour.
MessageChains	A class that uses a long chain of method invocations to realise (at least) one of its functionality.
RefusedParentBequest	A class that redefines inherited method using empty bodies, thus breaking polymorphism.
SpaghettiCode	A class with no structure, declaring long methods with no parameters, and utilising global variables. The names of the class and methods may suggest procedural programming. It does not exploit and prevents the use of object-orientation mechanisms, polymorphism, and inheritance.
SpeculativeGenerality	A class that is defined as abstract but that has very few children, which do not make use of its methods.
SwissArmyKnife	A complex class that offers a high number of services, for example, a complex class implementing a high number of interfaces. A Swiss Army Knife is different from a Blob, because it exposes a high complexity to address all foreseeable needs of a part of a system, whereas the Blob is a singleton monopolising all processing and data of a system. Thus, several Swiss Army Knives may exist in a system, for example utility classes.
TraditionBreaker	A class that inherits from a large parent class but that provides little behaviour and without subclasses.

Figure 9. Anti-pattern definitions

2.3.5 SQUAD

SQUAD is the short of Software Quality Understanding through the Analysis of Design. There are two sub models: QMOOD and PQMOD:

- QMOOD: evaluate the quality of the software systems including reusability, functionality, effectiveness, understandability and so on based on the outcome of POM analysis.
- PQMOD: It is based on the result of QMOOD but involve the design patterns factor. And the quality analysis changes to scalability, generality, modularity and etc.

The Figure 10 shows the entire structure of Ptidej tool suite services.

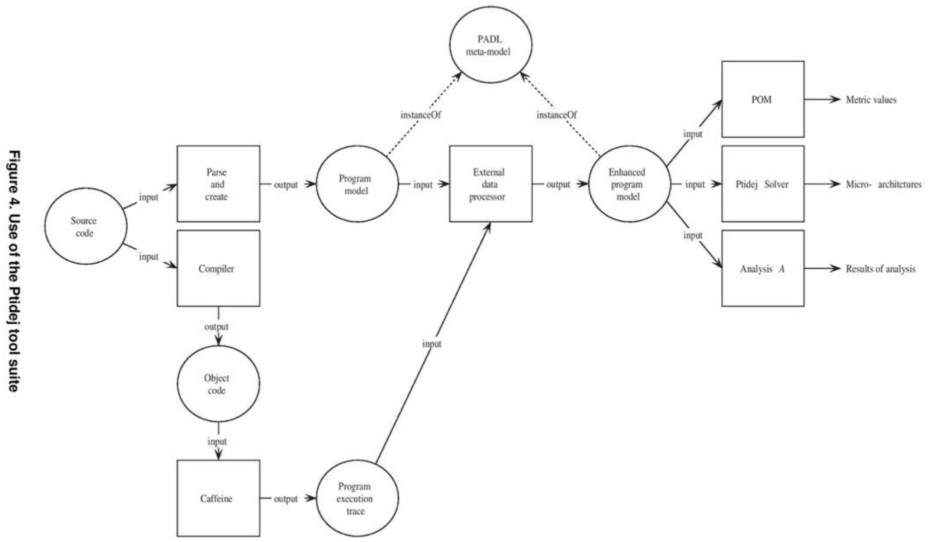


Figure 10. Use of Ptidej Tool Suite

The half above of Figure 10 shows the static analysis of Ptidej tool suite and the half below shows the dynamic analysis (CAFFEINE). Both static and dynamic analysis of the original source codes will produce the enhanced program model.

Chapter 3. Design Pattern

As we know, the Ptidej tool suite is to evaluate and enhance the quality of object-oriented programs by identifying the design patterns, the occurrences of design patterns in the models of programs. Therefore, the design pattern is the crucial part of Ptidej tool suite.

This chapter introduces some basic categories under design patterns.

3.1 Overview of Design Pattern

Design patterns represents the best practices used by experienced object-oriented software developers. Design patterns are solutions to general problems that software developers faced during software development. These solutions were obtained by trial and error by numerous software developers over quite a substantial period of time. [4]

There are two main usages of design patterns.

First of all, the design pattern is the standard methodology which can be used to solve a specific type of problem. What’s more, the design patterns are developed by the experienced developers and have been improved during a long time. Therefore, they are the best solutions to some software scenarios, which can be used by new developers to learn and make progress.

3.2 Types of Design Patterns

There are 23 kinds of design patterns and can be divided into 4 categories including Creational Patterns, Structural Patterns, Behavioral Patterns as well as J2EE Patterns.

3.2.1 Creational Patterns

Instead of producing an object directly from the new operator of a class, the creational patterns offer a method to create an object by hiding the inside creation logic, which gives the program more flexible to decide which object needs to be created for various scenarios.

There are some types of creational patterns including factory pattern, singleton pattern, builder pattern and etc. Here we just take singleton pattern as an example.

The singleton design pattern is the software design pattern which constrains the creation of an object by new operator of a class. This design pattern is used when there is only one object needed or restricting instantiating a certain number of objects though the software system. Here is a detailed example of implementation.

The Figure 11 shows the implementation of singleton design pattern. There is a static method of the SingleObject class to return its instance to the caller. SingletonPatternDemo is to get the object of SingleObject class.

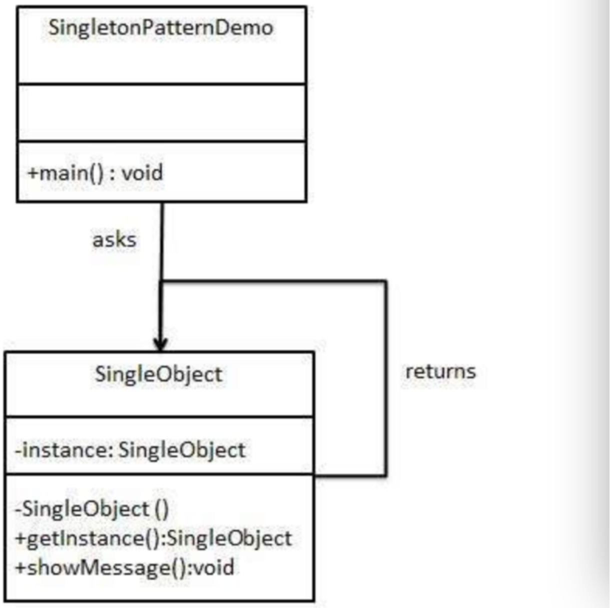
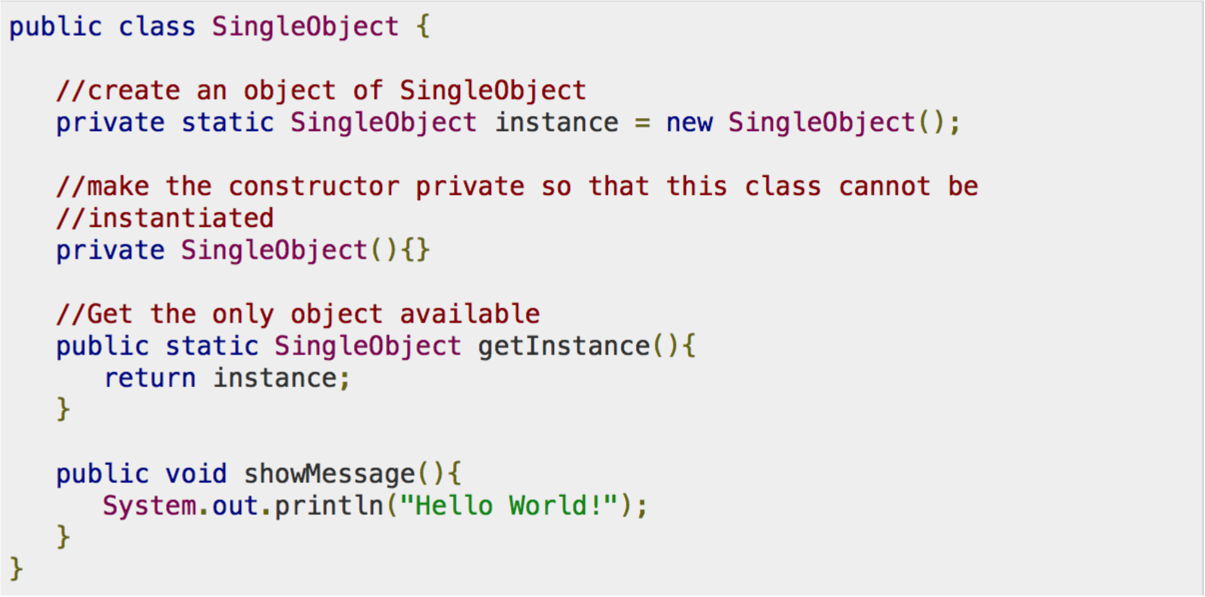


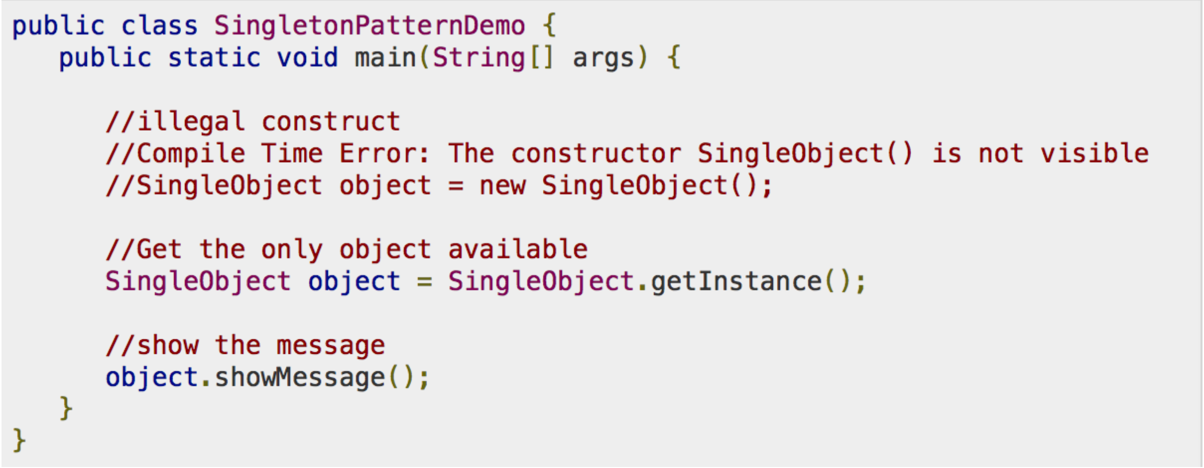
Figure 11. Implementation of Singleton Design Pattern

Figure 12 is to create a Singleton Class and Figure 13 is to obtain the object from the singleton class.



```java
public class SingleObject {

   //create an object of SingleObject
   private static SingleObject instance = new SingleObject();

   //make the constructor private so that this class cannot be
   //instantiated
   private SingleObject(){}

   //Get the only object available
   public static SingleObject getInstance(){
      return instance;
   }

   public void showMessage(){
      System.out.println("Hello World!");
   }
}
```

Figure 12. SingleObject.java



```java
public class SingletonPatternDemo {
   public static void main(String[] args) {

      //illegal construct
      //Compile Time Error: The constructor SingleObject() is not visible
      //SingleObject object = new SingleObject();

      //Get the only object available
      SingleObject object = SingleObject.getInstance();

      //show the message
      object.showMessage();
   }
}
```

Figure 13. SingletonPatternDemo.java

3.2.2 Structural Patterns

The structural patterns are mainly used for the composition of classes and objects. We can composite interfaces or different objects to get some new functionalities.

There are a few design patterns of structural patterns including composite patterns, façade patterns, proxy patterns and etc. Here we just take the composite pattern as an example.

The composite patterns are used when we consider different objects as a single object. The structural design patterns organize the objects in a tree structure to display different parts and the whole hierarchy as well.

The Figure 14 shows one implementation of structural design patterns.

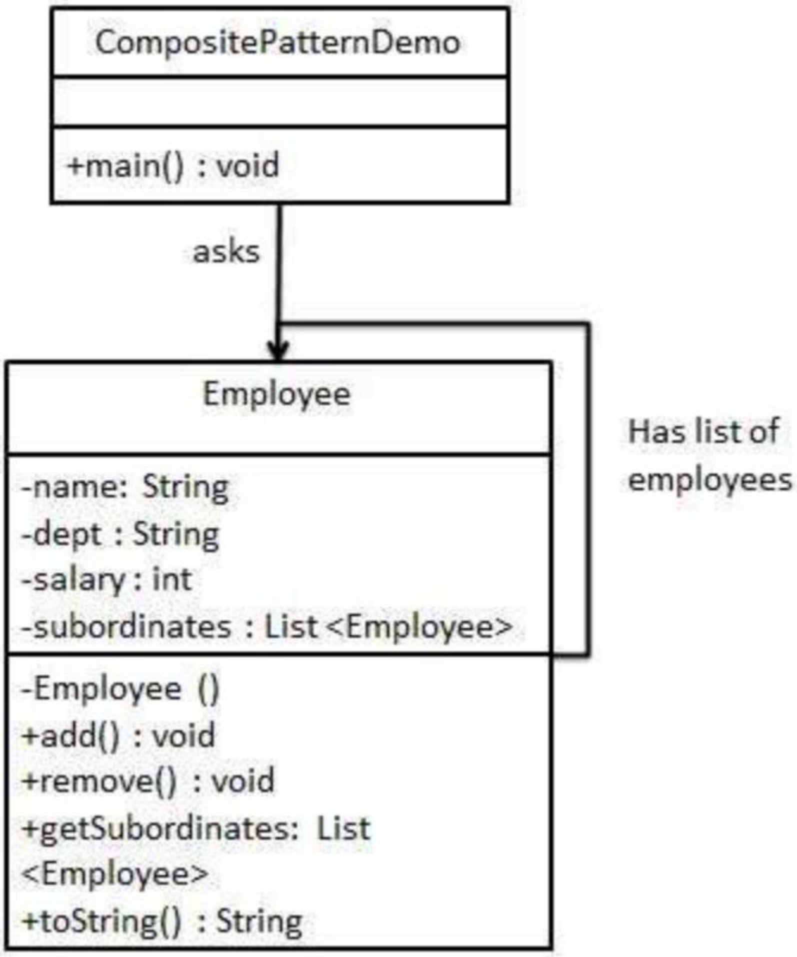
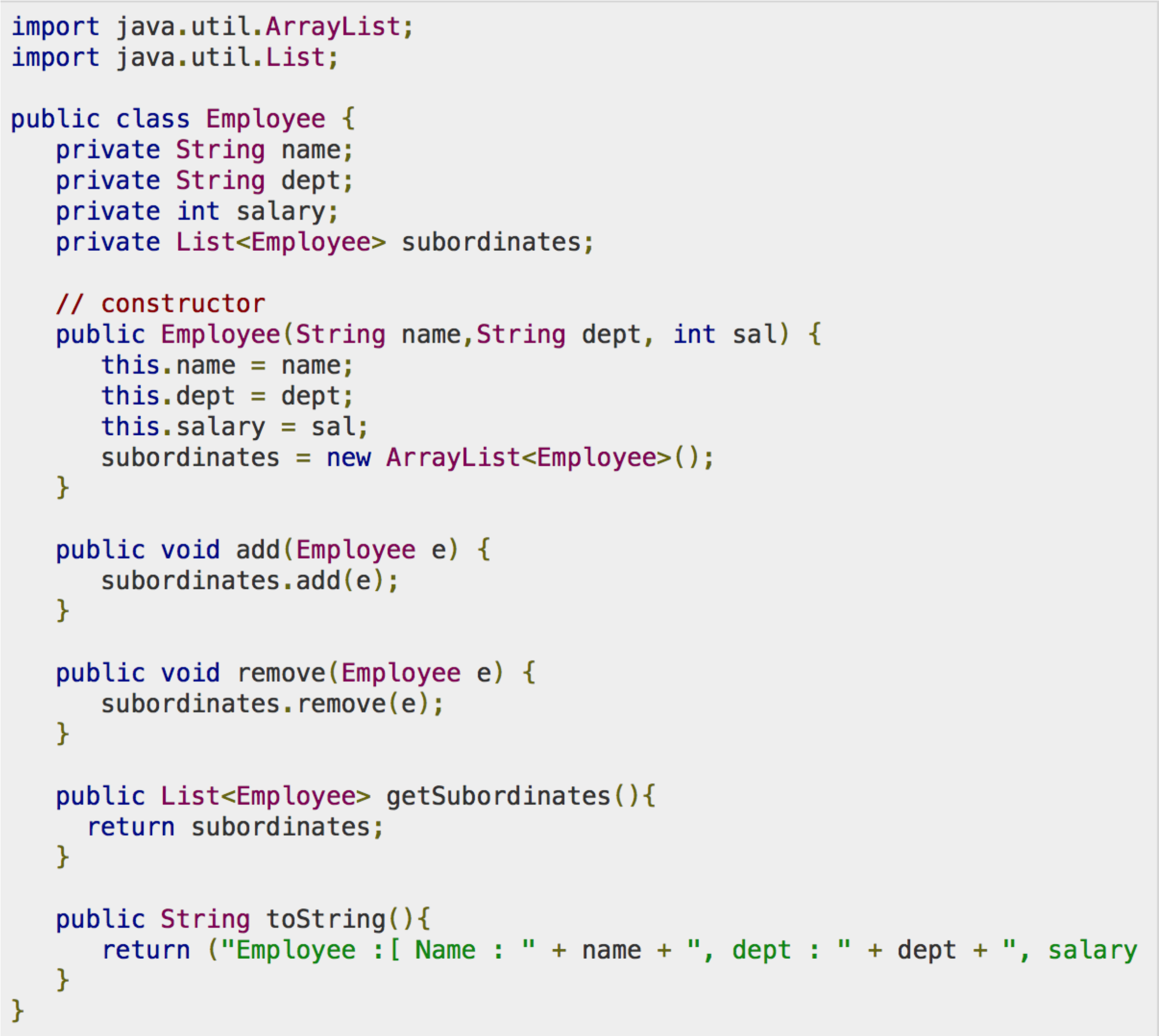


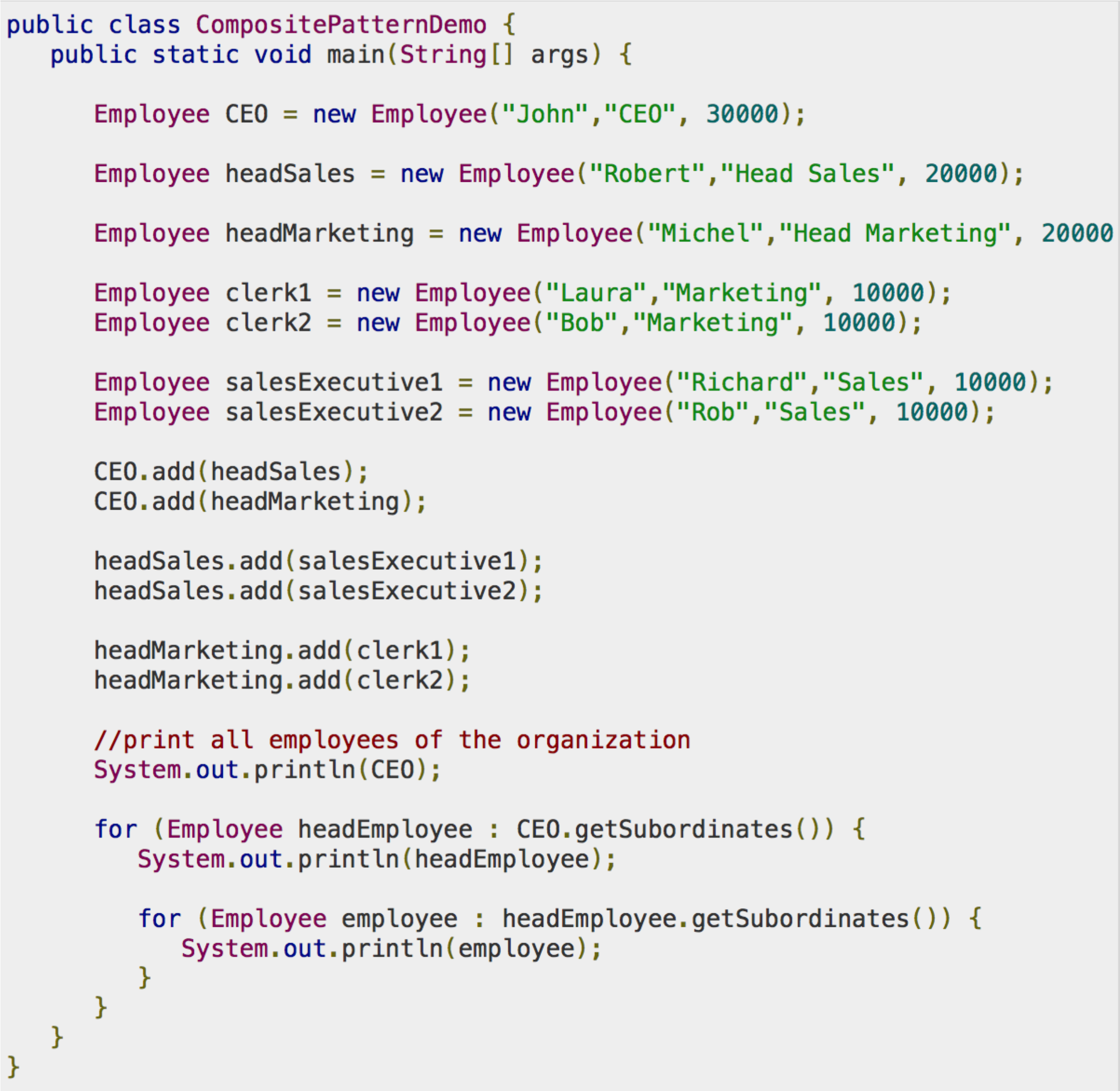
Figure 14. Implementation of Structural Design Patterns

The Figure 15 is to create Employee class containing many Employee objects. And the Figure 16 is to use the Employee class to generate the hierarchy of employees.



```java
import java.util.ArrayList;
import java.util.List;

public class Employee {
   private String name;
   private String dept;
   private int salary;
   private List<Employee> subordinates;

   // constructor
   public Employee(String name,String dept, int sal) {
      this.name = name;
      this.dept = dept;
      this.salary = sal;
      subordinates = new ArrayList<Employee>();
   }

   public void add(Employee e) {
      subordinates.add(e);
   }

   public void remove(Employee e) {
      subordinates.remove(e);
   }

   public List<Employee> getSubordinates(){
     return subordinates;
   }

   public String toString(){
      return ("Employee :[ Name : " + name + ", dept : " + dept + ", salary
   }
}
```

Figure 15. Employee.java



```java
public class CompositePatternDemo {
   public static void main(String[] args) {

      Employee CEO = new Employee("John","CEO", 30000);

      Employee headSales = new Employee("Robert","Head Sales", 20000);

      Employee headMarketing = new Employee("Michel","Head Marketing", 20000

      Employee clerk1 = new Employee("Laura","Marketing", 10000);
      Employee clerk2 = new Employee("Bob","Marketing", 10000);

      Employee salesExecutive1 = new Employee("Richard","Sales", 10000);
      Employee salesExecutive2 = new Employee("Rob","Sales", 10000);

      CEO.add(headSales);
      CEO.add(headMarketing);

      headSales.add(salesExecutive1);
      headSales.add(salesExecutive2);

      headMarketing.add(clerk1);
      headMarketing.add(clerk2);

      //print all employees of the organization
      System.out.println(CEO);

      for (Employee headEmployee : CEO.getSubordinates()) {
         System.out.println(headEmployee);

         for (Employee employee : headEmployee.getSubordinates()) {
            System.out.println(employee);
         }
      }
   }
}
```

Figure 16. CompositePatternDemo.java

3.2.3 Behavioral Patterns

The behavioral design patterns mainly focus on the communication among different objects/classes. There are a few kinds of behavioral design patterns including the chain of responsibility pattern, command pattern, interpreter pattern and etc. Here we take the chain of responsibility as an example.

In the behavioral design patterns, there are a sequence of receivers to solve a request. If the object cannot solve the request, it will pass this request to the next receiver. The Figure 17 shows the implementation of chain of responsibility design patterns.

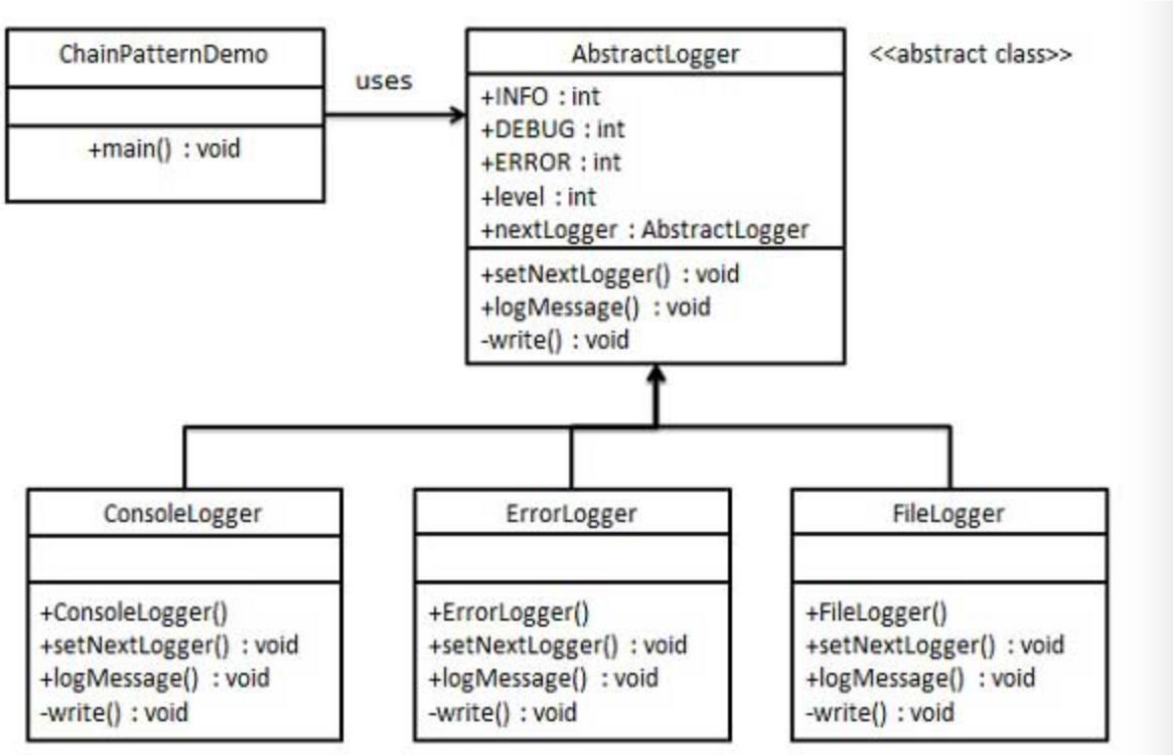
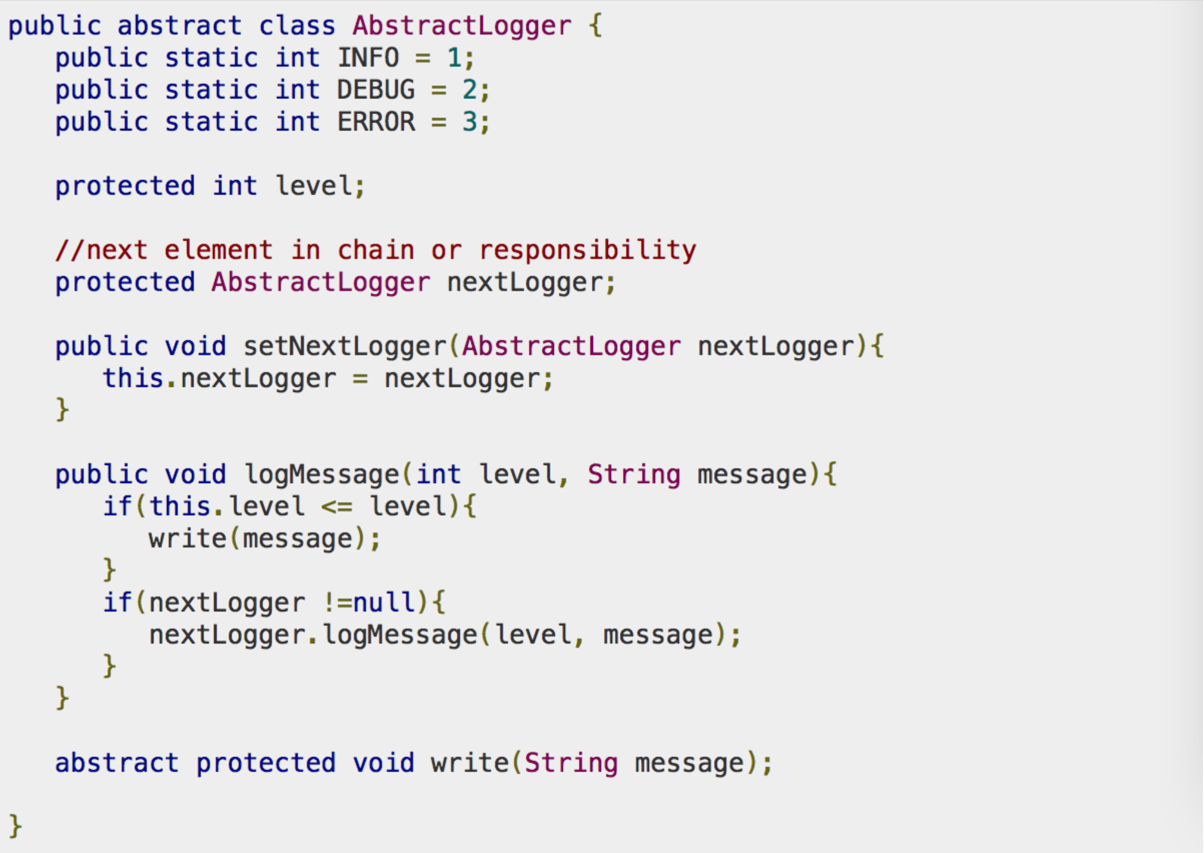


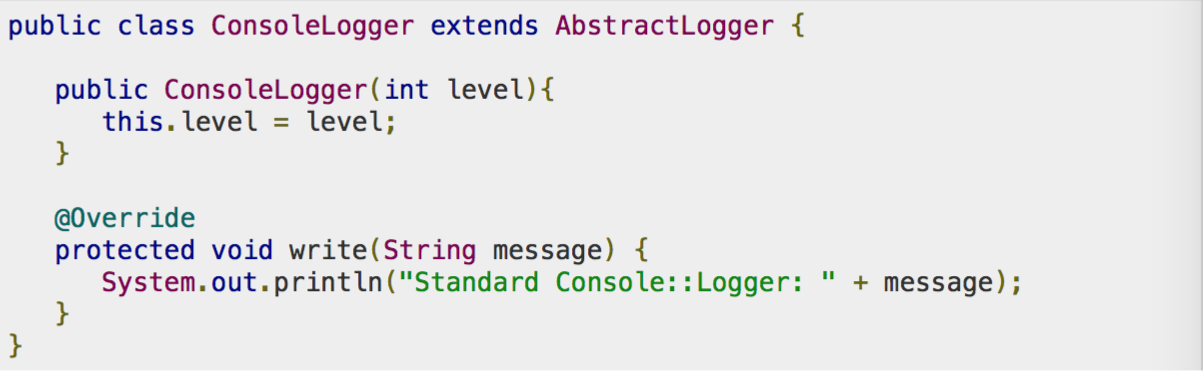
Figure 17. Implementation of Chain of Responsibility

The Figure 18 is to create an abstract logger class. The Figure 19, Figure 20 and Figure 21 are to concrete classes extending the logger. Figure 22 is to produce various type of loggers and put the next logger reference in each logger, which stands for the chain in the chain of responsibility design patterns.



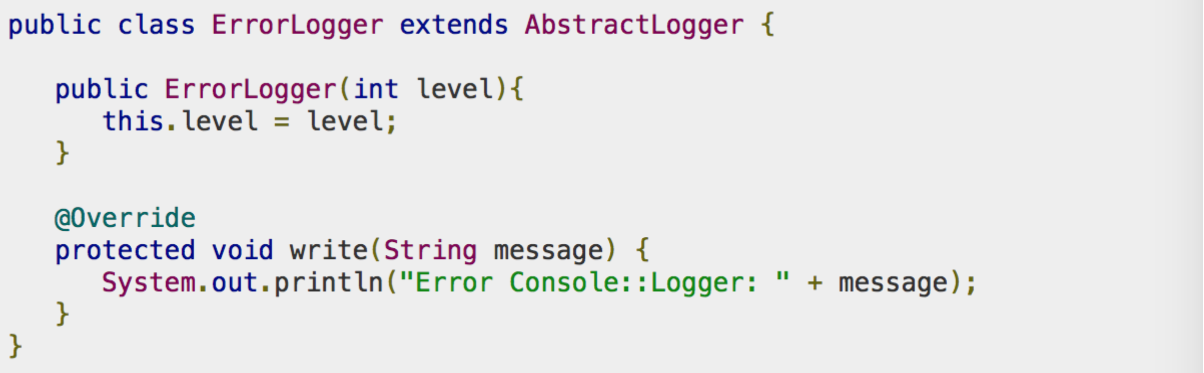
```java
public abstract class AbstractLogger {
   public static int INFO = 1;
   public static int DEBUG = 2;
   public static int ERROR = 3;

   protected int level;

   //next element in chain or responsibility
   protected AbstractLogger nextLogger;

   public void setNextLogger(AbstractLogger nextLogger){
      this.nextLogger = nextLogger;
   }

   public void logMessage(int level, String message){
      if(this.level <= level){
         write(message);
      }
      if(nextLogger !=null){
         nextLogger.logMessage(level, message);
      }
   }

   abstract protected void write(String message);

}
```

Figure 18. AbstractLogger.java



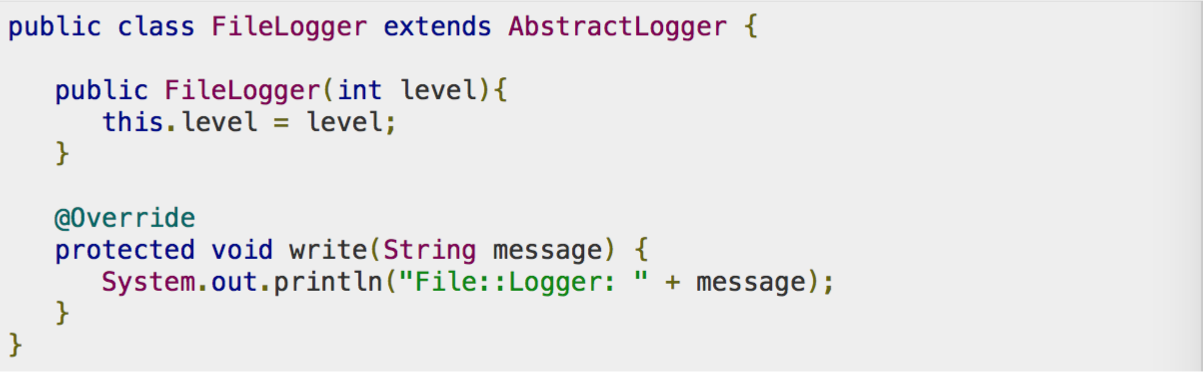
```java
public class ConsoleLogger extends AbstractLogger {

   public ConsoleLogger(int level){
      this.level = level;
   }

   @Override
   protected void write(String message) {
      System.out.println("Standard Console::Logger: " + message);
   }
}
```

Figure 19. ConsoleLogger.java



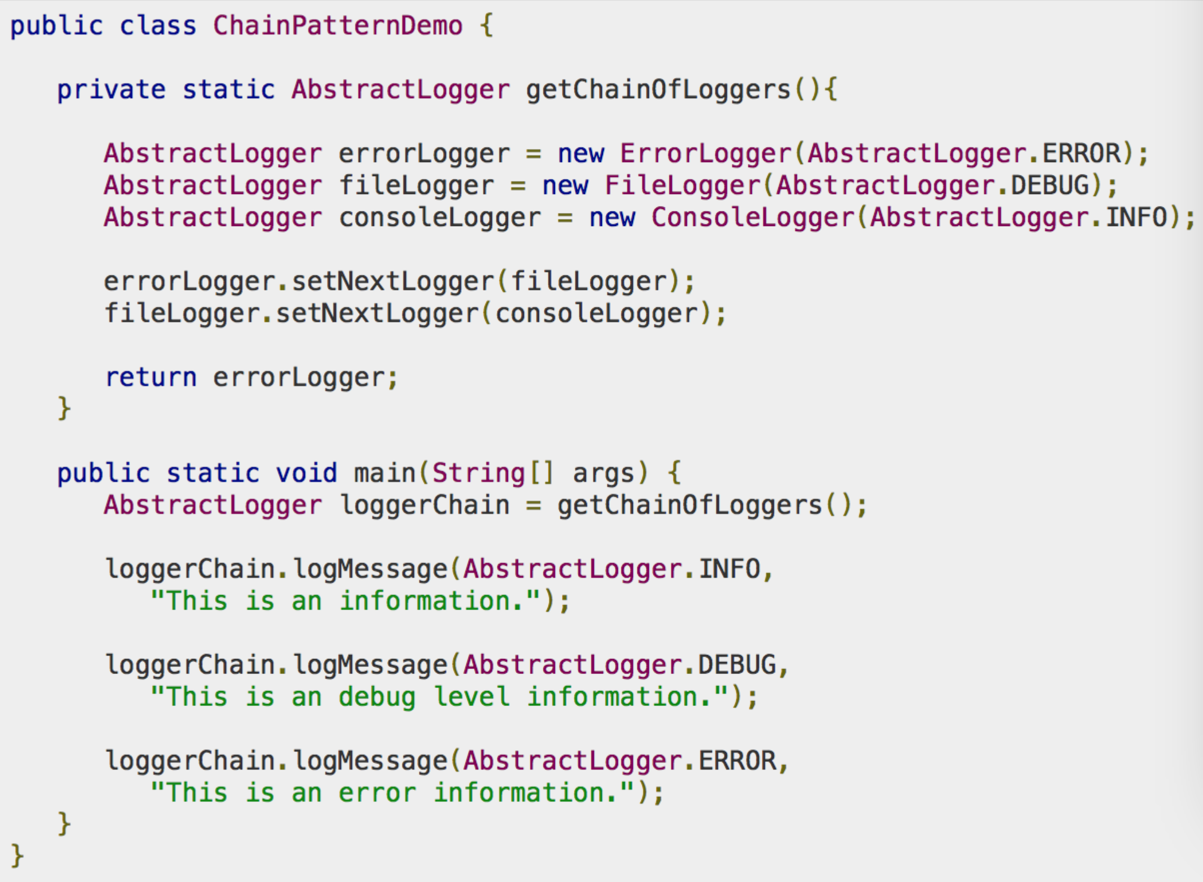
```java
public class ErrorLogger extends AbstractLogger {

   public ErrorLogger(int level){
      this.level = level;
   }

   @Override
   protected void write(String message) {
      System.out.println("Error Console::Logger: " + message);
   }
}
```

Figure 20. ErrorLogger.java



```java
public class FileLogger extends AbstractLogger {

   public FileLogger(int level){
      this.level = level;
   }

   @Override
   protected void write(String message) {
      System.out.println("File::Logger: " + message);
   }
}
```

Figure 21. FileLogger.java



```java
public class ChainPatternDemo {

   private static AbstractLogger getChainOfLoggers(){

      AbstractLogger errorLogger = new ErrorLogger(AbstractLogger.ERROR);
      AbstractLogger fileLogger = new FileLogger(AbstractLogger.DEBUG);
      AbstractLogger consoleLogger = new ConsoleLogger(AbstractLogger.INFO);

      errorLogger.setNextLogger(fileLogger);
      fileLogger.setNextLogger(consoleLogger);

      return errorLogger;
   }

   public static void main(String[] args) {
      AbstractLogger loggerChain = getChainOfLoggers();

      loggerChain.logMessage(AbstractLogger.INFO,
         "This is an information.");

      loggerChain.logMessage(AbstractLogger.DEBUG,
         "This is an debug level information.");

      loggerChain.logMessage(AbstractLogger.ERROR,
         "This is an error information.");
   }
}
```

Figure 22. ChainPatternDemo.java

3.2.4 J2EE Patterns

The J2EE design patterns are dealing with the presentation tier typically. And Sun Java Center identifies the J2EE design patterns.

Chapter 4. Configure the Glassfish Server

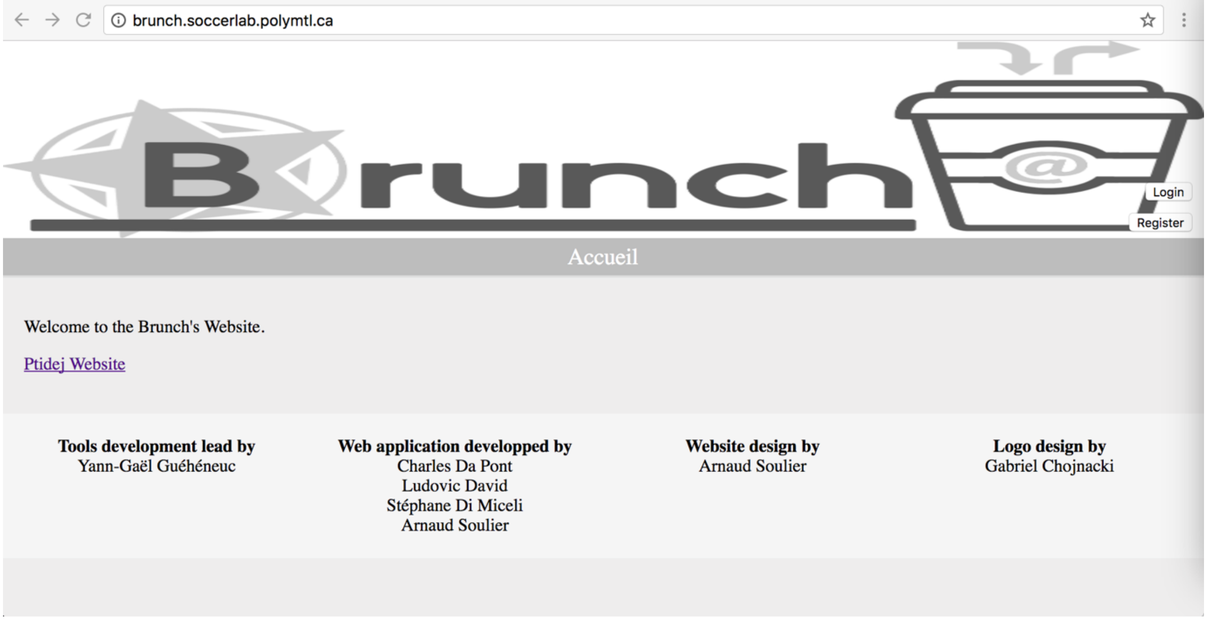
From the content above, we know that Ptidej tool suite is to evaluate and to enhance the quality of software systems especially object-oriented programs by identifying the design patterns and the occurrence of design patterns.

Moreover, there is a online version of Ptidej tool suite called brunch. The official website of "brunch" is http://brunch.soccerlab.polymtl.ca/. And we can download the source codes of "brunch" directly from bitbucket https://bitbucket.org/ptidejteam/brunch.

4.1 Introduction to "brunch" website

The "brunch" website is the online version of the Ptidej tool suite. You can select your .jar file and input to the website, "brunch" will help to perform some services of Ptidej tool suite including Ptidej Solver, SAD, POM, SQUAD and etc. The details will be provided below.

This is the "brunch" website user interface at the very beginning.



Welcome to the Brunch's Website.

Ptidej Website

Tools development lead by
Yann-Gaël Guéhéneuc

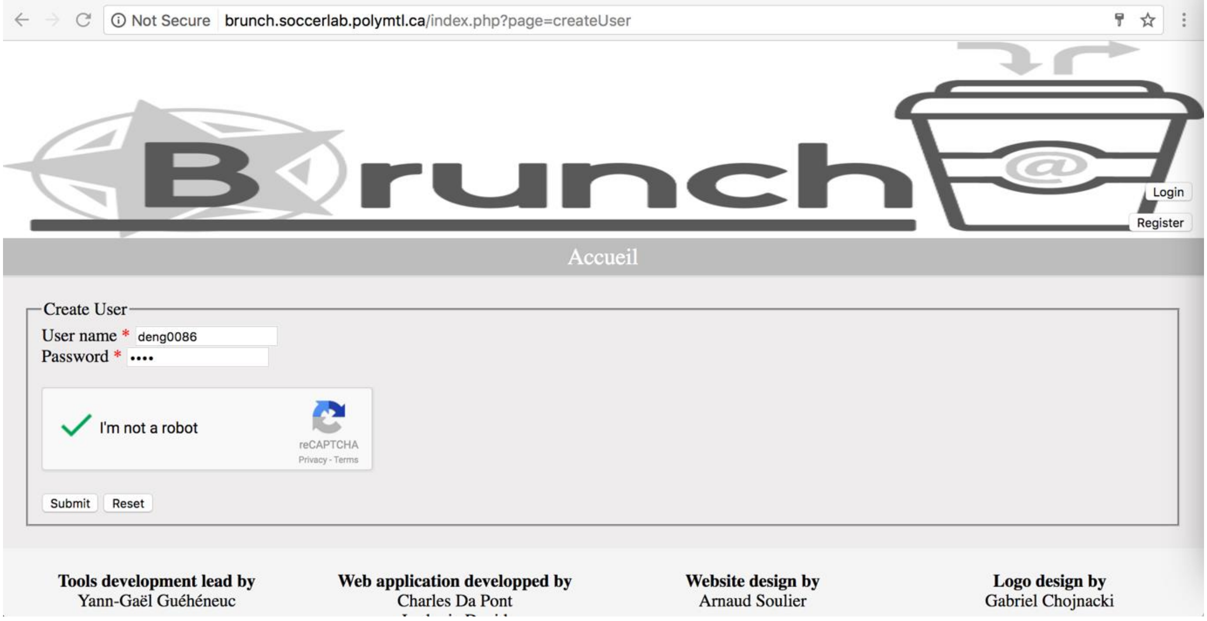
Web application developped by
Charles Da Pont
Ludovic David
Stéphane Di Miceli
Arnaud Soulier

Website design by
Arnaud Soulier

Logo design by
Gabriel Chojnacki

Figure 23. The user interface of brunch at the beginning

First of all, you need to register a user by clicking the "Register" button on the right column.



Not Secure brunch.soccerlab.polymtl.ca/index.php?page=createUser
Brunch
Login
Register
Accueil

Create User

User name * deng0086
Password * ••••

☑ I'm not a robot

reCAPTCHA
Privacy - Terms
Submit
Reset

Tools development lead by
Yann-Gaël Guéhéneuc

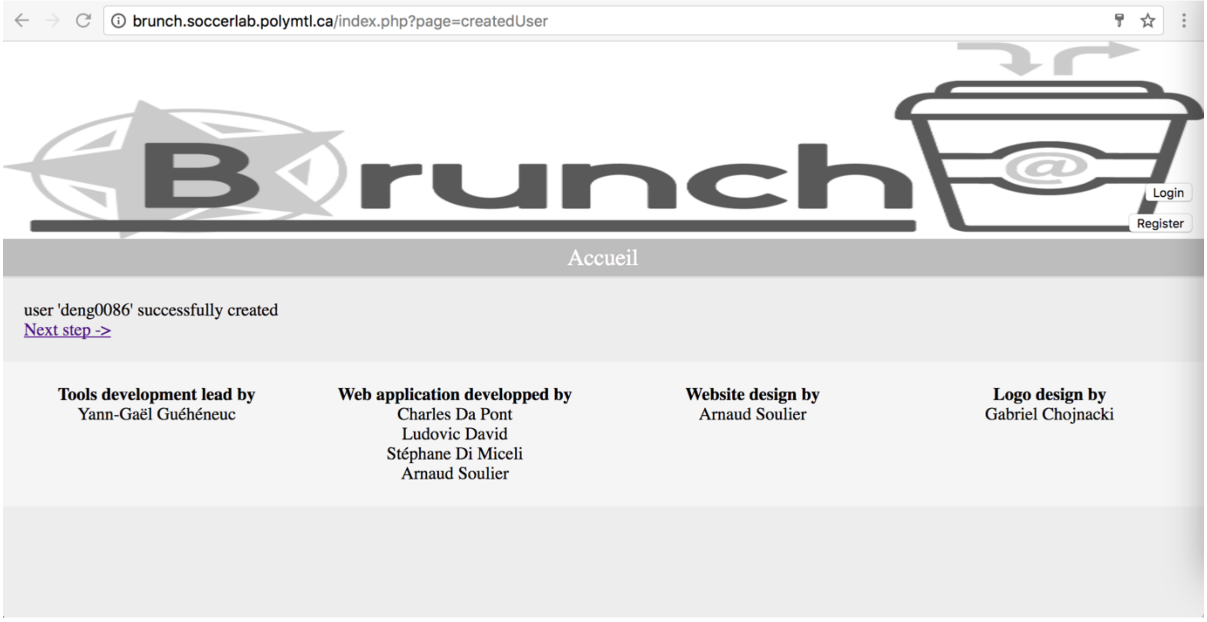
Web application developped by
Charles Da Pont

Website design by
Arnaud Soulier

Logo design by
Gabriel Chojnacki

Figure 24. Register a user account at "brunch"

After type in your username and password, pass the google reCAPTCHA, click the “Submit” button to process to the next step.



brunch.soccerlab.polymtl.ca/index.php?page=createdUser

user 'deng0086' successfully created
Next step ->

Tools development lead by
Yann-Gaël Guéhéneuc

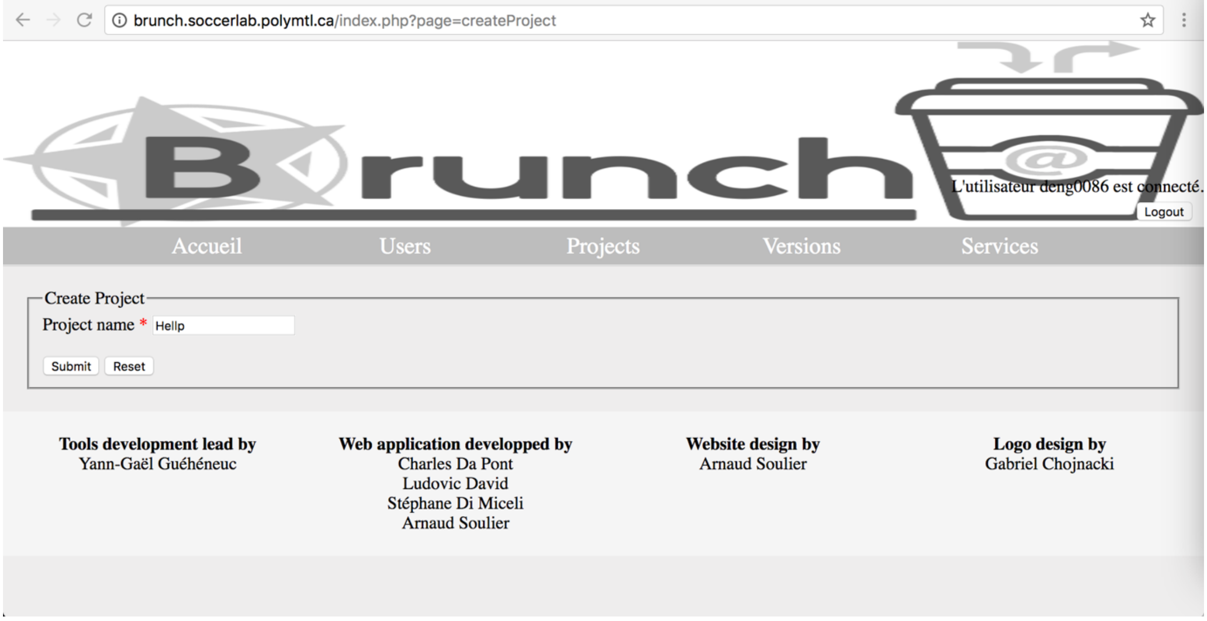
Web application developped by
Charles Da Pont
Ludovic David
Stéphane Di Miceli
Arnaud Soulier

Website design by
Arnaud Soulier

Logo design by
Gabriel Chojnacki

Figure 25. user account successfully created

After click “Next step”, you can choose your project name.



brunch.soccerlab.polymtl.ca/index.php?page=createProject

L'utilisateur deng0086 est connecté.

Create Project
Project name * Hellp

Submit Reset

Tools development lead by
Yann-Gaël Guéhéneuc

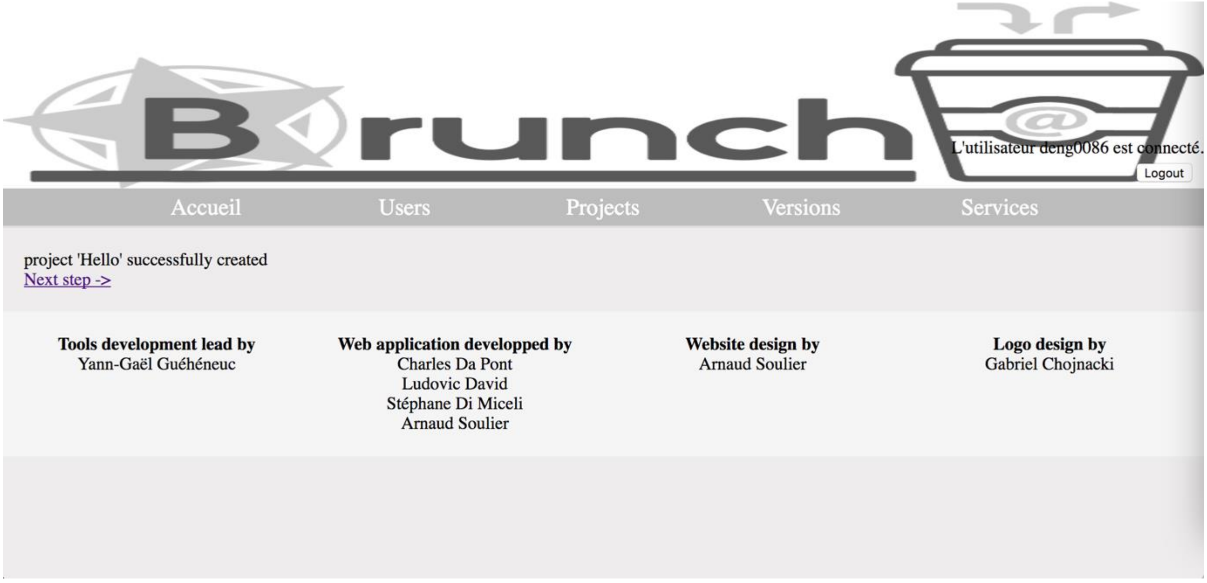
Web application developped by
Charles Da Pont
Ludovic David
Stéphane Di Miceli
Arnaud Soulier

Website design by
Arnaud Soulier

Logo design by
Gabriel Chojnacki

Figure 26. create project at “brunch”

After type in the project name chosen and click “Submit”, you can process to the next step.



L'utilisateur deng0086 est connecté.

project 'Hello' successfully created
Next step ->

Tools development lead by
Yann-Gaël Guéhéneuc

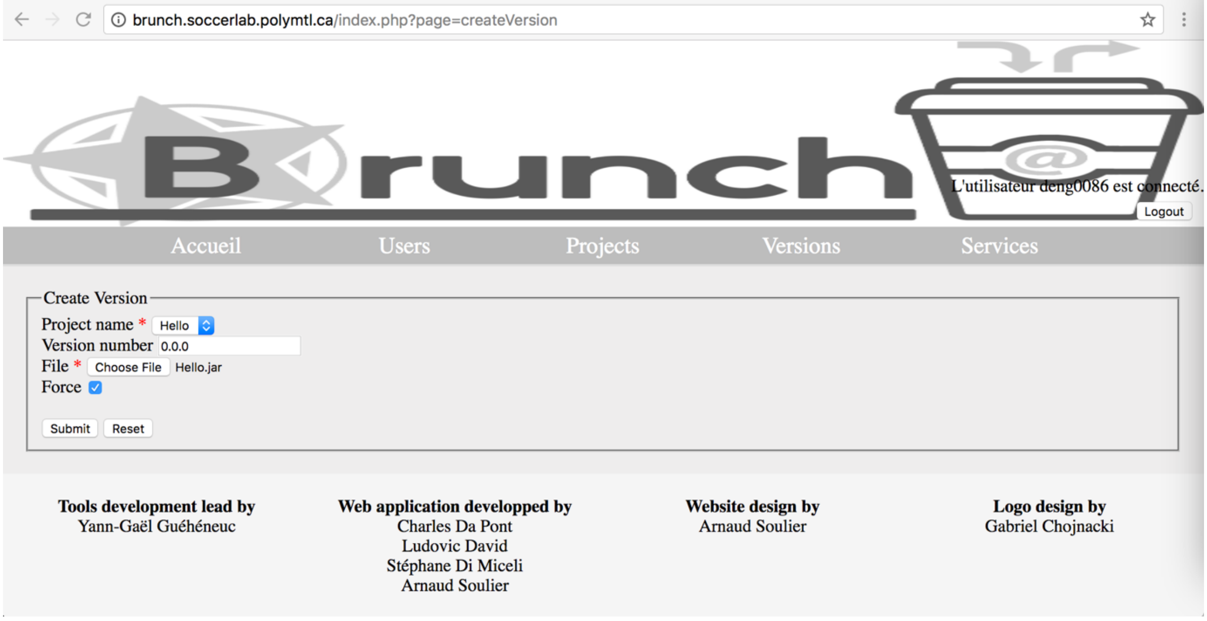
Web application developped by
Charles Da Pont
Ludovic David
Stéphane Di Miceli
Arnaud Soulier

Website design by
Arnaud Soulier

Logo design by
Gabriel Chojnacki

Figure 27. project name successfully created

After clicking "Next step", you are directed to the "Create Version" page.



brunch.soccerlab.polymtl.ca/index.php?page=createVersion

L'utilisateur deng0086 est connecté.

Create Version
Project name * Hello
Version number 0.0.0
File * Hello.jar
Force ☑

Tools development lead by
Yann-Gaël Guéhéneuc

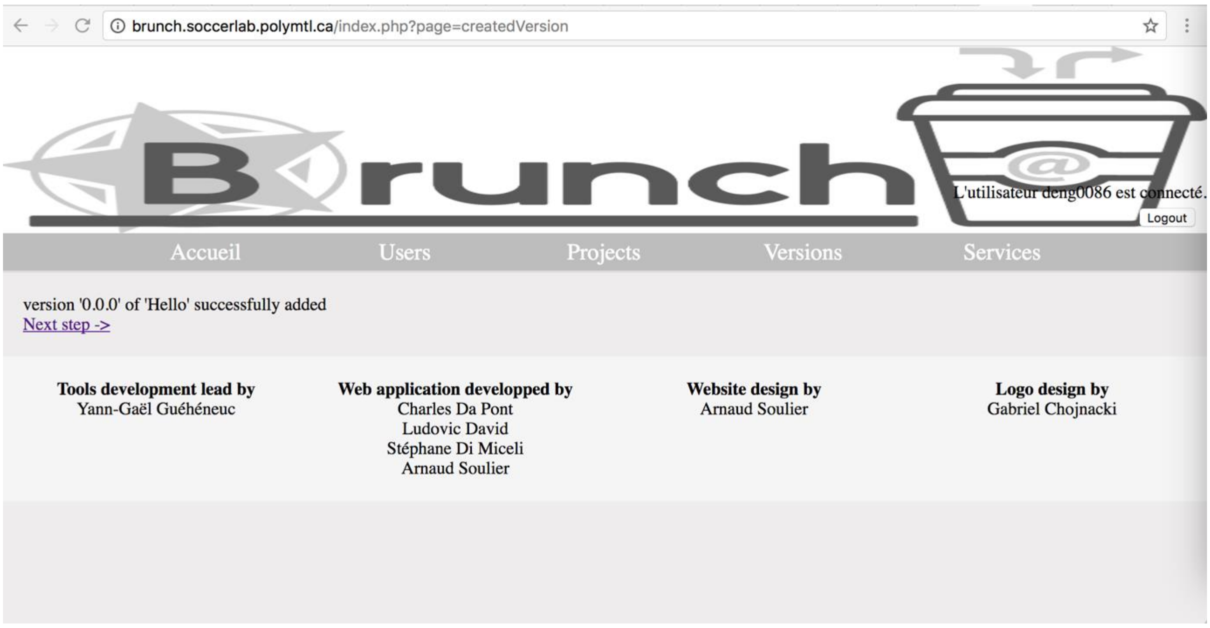
Web application developped by
Charles Da Pont
Ludovic David
Stéphane Di Miceli
Arnaud Soulier

Website design by
Arnaud Soulier

Logo design by
Gabriel Chojnacki

Figure 28. create version at brunch

After select the project name, version number and choose the file name, Hello.jar, then you can click "Submit" to process to the next page.


brunch.soccerlab.polymtl.ca/index.php?page=createdVersion
Brunch
L'utilisateur deng0086 est connecté.
Logout
Accueil
Users
Projects
Versions
Services

version '0.0.0' of 'Hello' successfully added
Next step ->

Tools development lead by
Yann-Gaël Guéhéneuc

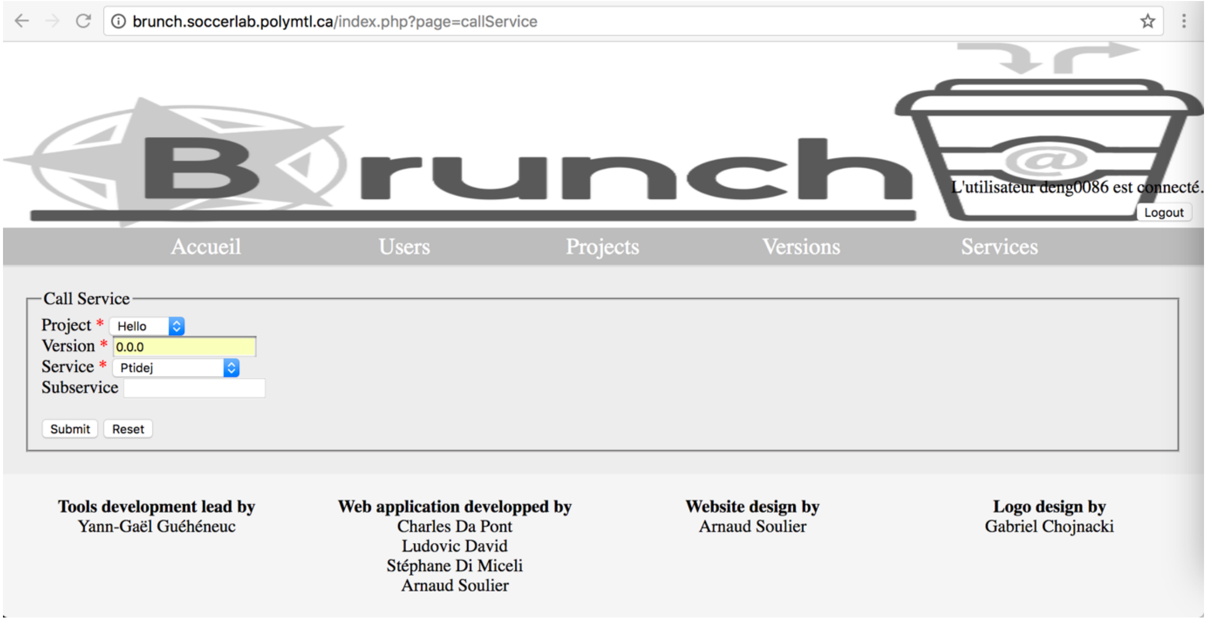
Web application developped by
Charles Da Pont
Ludovic David
Stéphane Di Miceli
Arnaud Soulier

Website design by
Arnaud Soulier

Logo design by
Gabriel Chojnacki

Figure 29. version successfully created

After click “Next step”, you can go to the select the services.


brunch.soccerlab.polymtl.ca/index.php?page=callService
Brunch
L'utilisateur deng0086 est connecté.
Logout
Accueil
Users
Projects
Versions
Services
Submit
Reset

Call Service
Project * Hello
Version * 0.0.0
Service * Ptidej
Subservice

Tools development lead by
Yann-Gaël Guéhéneuc

Web application developped by
Charles Da Pont
Ludovic David
Stéphane Di Miceli
Arnaud Soulier

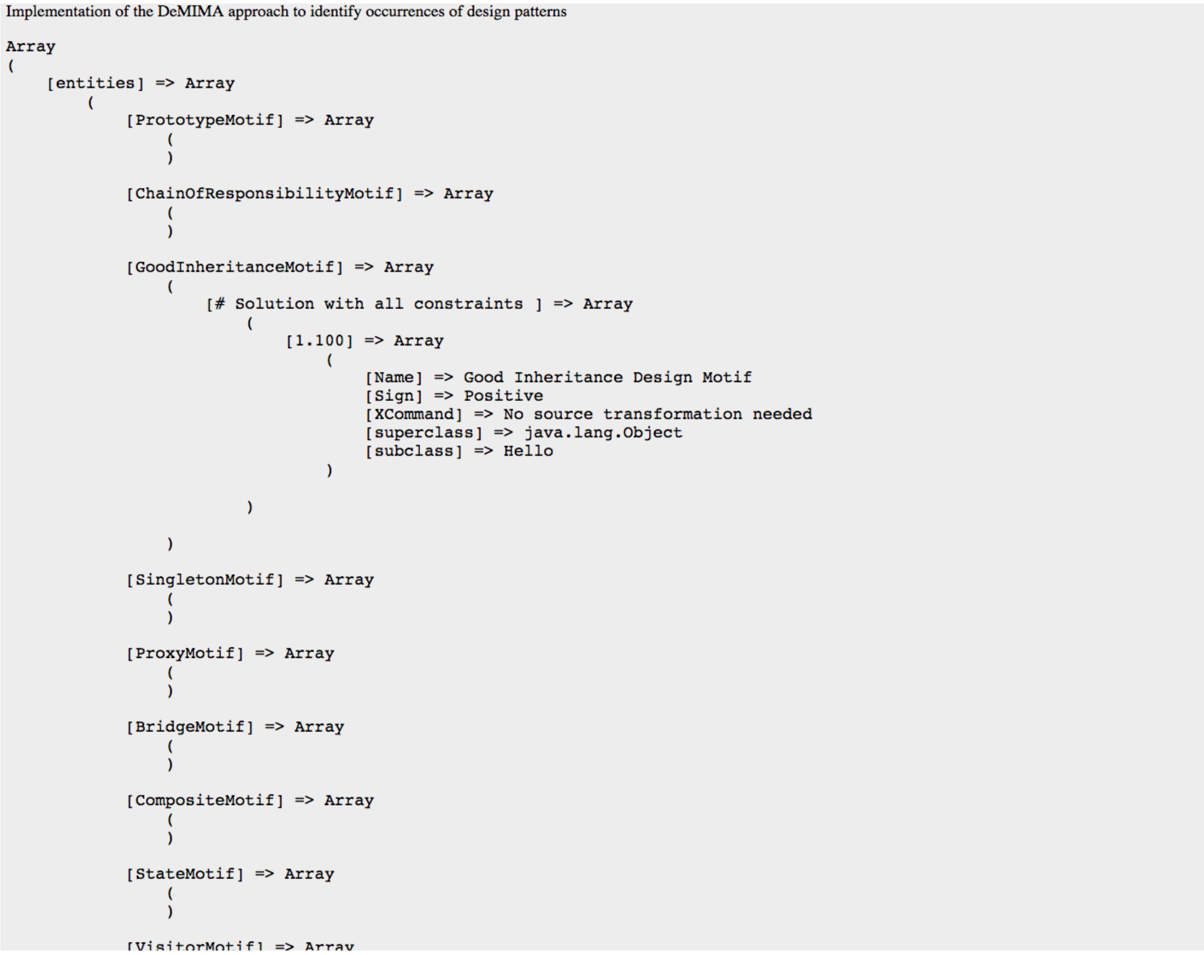
Website design by
Arnaud Soulier

Logo design by
Gabriel Chojnacki

Figure 30. call service at brunch

There are totally 6 services you can select: Ptidej Solver, Stats, POM, SAD, SQUAD/PQMOD, SQUAD/QMOOD.

The Ptidej Solver can help to identify the design patterns as well as the roles different classes play in the design pattern. For example, in the Hello.jar model, a design pattern called “GoodInheritanceMotif” is identified and the “Hello” class play a role of “subclass in this design pattern.

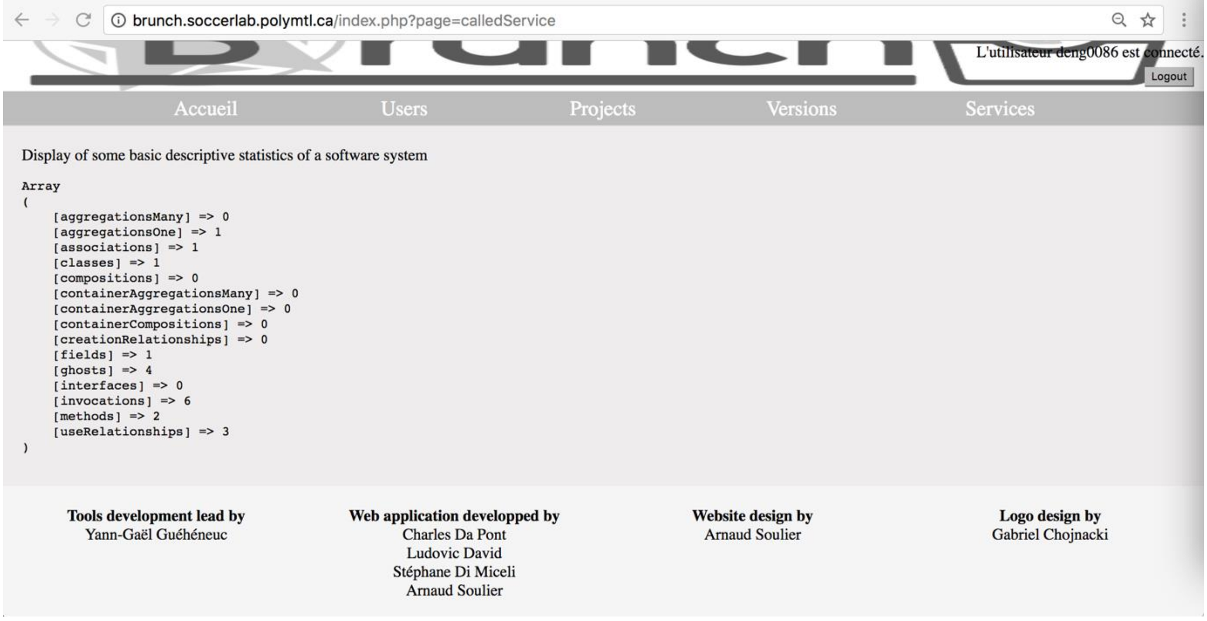


Implementation of the DeMIMA approach to identify occurrences of design patterns

```
Array
(
    [entities] => Array
        (
            [PrototypeMotif] => Array
                (
                )

            [ChainOfResponsibilityMotif] => Array
                (
                )

            [GoodInheritanceMotif] => Array
                (
                    [# Solution with all constraints ] => Array
                        (
                            [1.100] => Array
                                (
                                    [Name] => Good Inheritance Design Motif
                                    [Sign] => Positive
                                    [XCommand] => No source transformation needed
                                    [superclass] => java.lang.Object
                                    [subclass] => Hello
                                )

                        )

                )

            [SingletonMotif] => Array
                (
                )

            [ProxyMotif] => Array
                (
                )

            [BridgeMotif] => Array
                (
                )

            [CompositeMotif] => Array
                (
                )

            [StateMotif] => Array
                (
                )

            [VisitorMotif] => Array
```

Figure 31. Ptidej Solver service at brunch

The Stats service can apply a few calculations on the model of program and return you the numerical numbers of classes, compositions, interfaces, methods and etc.


brunch.soccerlab.polymtl.ca/index.php?page=calledService
L'utilisateur deng0086 est connecté.
Logout
Accueil
Users
Projects
Versions
Services

Display of some basic descriptive statistics of a software system

```
Array
(
    [aggregationsMany] => 0
    [aggregationsOne] => 1
    [associations] => 1
    [classes] => 1
    [compositions] => 0
    [containerAggregationsMany] => 0
    [containerAggregationsOne] => 0
    [containerCompositions] => 0
    [creationRelationships] => 0
    [fields] => 1
    [ghosts] => 4
    [interfaces] => 0
    [invocations] => 6
    [methods] => 2
    [useRelationships] => 3
)
```

Tools development lead by
Yann-Gaël Guéhéneuc

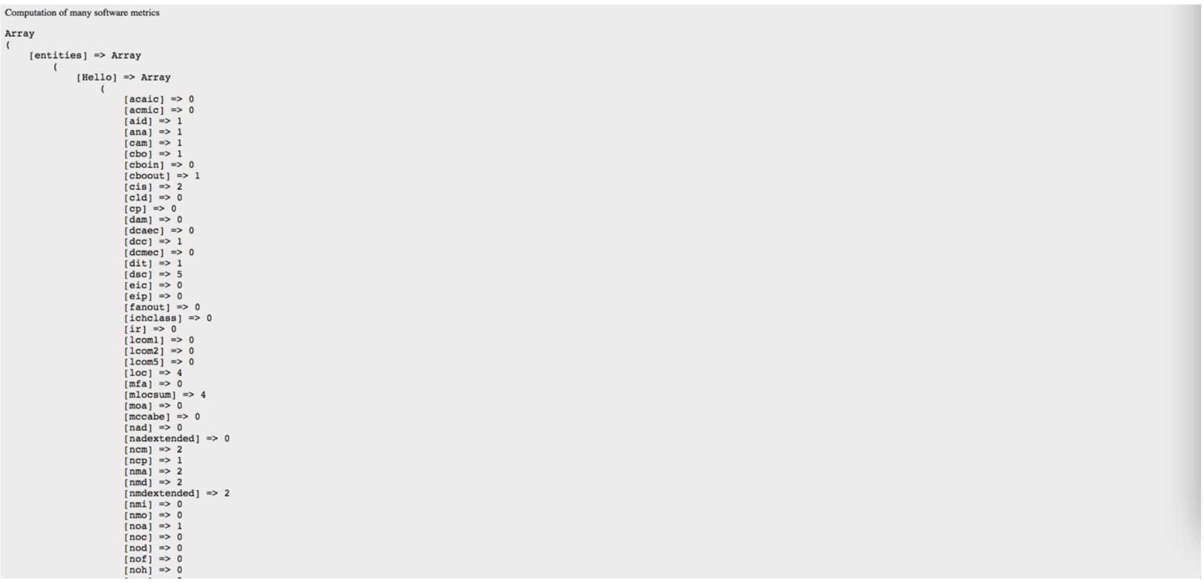
Web application developped by
Charles Da Pont
Ludovic David
Stéphane Di Miceli
Arnaud Soulier

Website design by
Arnaud Soulier

Logo design by
Gabriel Chojnacki

Figure 32. Stats Service at brunch

The POM service will apply a few metrics including AID, CIS, DIT and so on on the model of program. Some numerical values will return to you.



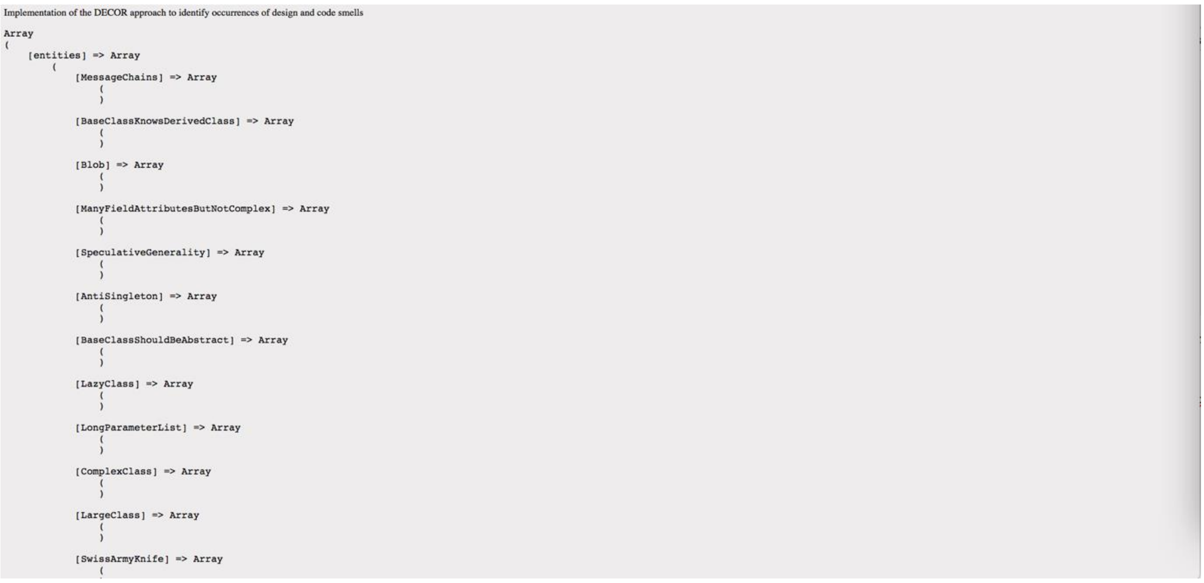
Computation of many software metrics

```
Array
(
    [entities] => Array
        (
            [Hello] => Array
                (
                    [acaic] => 0
                    [acmic] => 0
                    [aid] => 1
                    [ana] => 1
                    [cam] => 1
                    [cbo] => 1
                    [cboin] => 0
                    [cboout] => 1
                    [cis] => 2
                    [cld] => 0
                    [cp] => 0
                    [dam] => 0
                    [dcaec] => 0
                    [dcc] => 1
                    [dcmec] => 0
                    [dit] => 1
                    [dsc] => 5
                    [eic] => 0
                    [eip] => 0
                    [fanout] => 0
                    [ichclass] => 0
                    [ir] => 0
                    [lcom1] => 0
                    [lcom2] => 0
                    [lcom5] => 0
                    [loc] => 4
                    [mfa] => 0
                    [mlocsum] => 4
                    [moa] => 0
                    [mccabe] => 0
                    [nad] => 0
                    [nadextended] => 0
                    [ncm] => 2
                    [ncp] => 1
                    [nma] => 2
                    [nmd] => 2
                    [nmdextended] => 2
                    [nmi] => 0
                    [nmo] => 0
                    [noa] => 1
                    [noc] => 0
                    [nod] => 0
                    [nof] => 0
                    [noh] => 0
```

Figure 33. POM service at brunch

From the example above, we can see that in "Hello.jar", the AID (Average Inheritance of Depth) is 1; the CIS (Number of public methods in a class) is 2; the DIT (Depth of Inheritance Tree of an entity) is 1 and etc.

The SAD service is to identify the anti-design pattern including LazyClass, ComplexClass, Blob and etc.

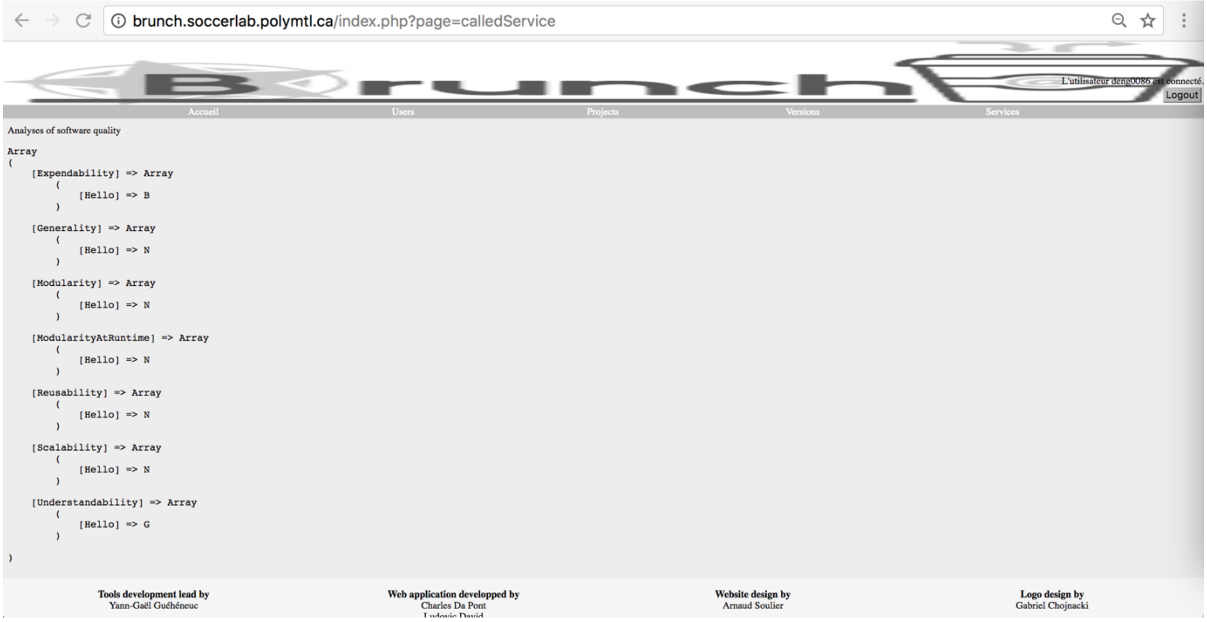


Implementation of the DECOR approach to identify occurrences of design and code smells

```
Array
(
    [entities] => Array
        (
            [MessageChains] => Array
                (
                )

            [BaseClassKnowsDerivedClass] => Array
                (
                )

            [Blob] => Array
                (
                )

            [ManyFieldAttributesButNotComplex] => Array
                (
                )

            [SpeculativeGenerality] => Array
                (
                )

            [AntiSingleton] => Array
                (
                )

            [BaseClassShouldBeAbstract] => Array
                (
                )

            [LazyClass] => Array
                (
                )

            [LongParameterList] => Array
                (
                )

            [ComplexClass] => Array
                (
                )

            [LargeClass] => Array
                (
                )

            [SwissArmyKnife] => Array
                (
```

Figure 34. SAD service at brunch

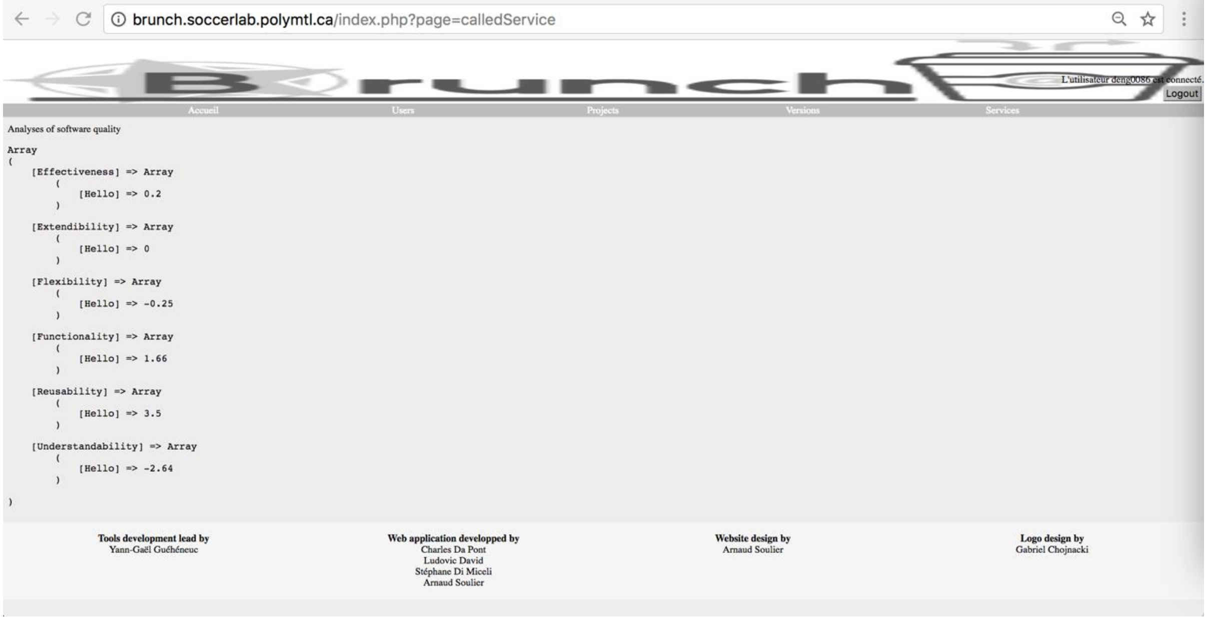
SQUAD/PQMOD and SQUAD/QMOOD are services to evaluate the quality of a software system including expendability, generality, modularity, reusability, understandability and etc. based on the result of POM analysis.


brunch.soccerlab.polymtl.ca/index.php?page=calledService
L'utilisateur deng0086 est connecté.
Logout
Accueil
Users
Projects
Versions
Services

Analyses of software quality

```
Array
(
    [Expendability] => Array
        (
            [Hello] => B
        )

    [Generality] => Array
        (
            [Hello] => N
        )

    [Modularity] => Array
        (
            [Hello] => N
        )

    [ModularityAtRuntime] => Array
        (
            [Hello] => N
        )

    [Reusability] => Array
        (
            [Hello] => N
        )

    [Scalability] => Array
        (
            [Hello] => N
        )

    [Understandability] => Array
        (
            [Hello] => G
        )

)
```

Figure 35. SQUAD/PQMOD Service at brunch


brunch.soccerlab.polymtl.ca/index.php?page=calledService
L'utilisateur deng0086 est connecté.
Logout
Accueil
Users
Projects
Versions
Services

Analyses of software quality

```
Array
(
    [Effectiveness] => Array
        (
            [Hello] => 0.2
        )

    [Extendibility] => Array
        (
            [Hello] => 0
        )

    [Flexibility] => Array
        (
            [Hello] => -0.25
        )

    [Functionality] => Array
        (
            [Hello] => 1.66
        )

    [Reusability] => Array
        (
            [Hello] => 3.5
        )

    [Understandability] => Array
        (
            [Hello] => -2.64
        )

)
```

Figure 36. SQUAD/QMOOD Service at brunch

For example, the functionality of Hello.jar is 1.66 and the reusability is 3.5.

These are all basic work flows of brunch website.

4.2 GlassFish Server

My internship work is basically to improve the brunch website by modifying some PHP source codes, therefore, I need to run the brunch web application on my localhost at the beginning.

GlassFish Server provides a server for the development and deployment of Java Platform, Enterprise Edition (Java EE platform) applications and web technologies based on Java technology. GlassFish Server 4.1 provides the following:

- A lightweight and extensible core based on OSGI Alliance standards

- A web container
- An easy-to-use Administration Console for configuration and management
- Update Tool connectivity for updates and add-on components [5]

After downloading the “brunch” source codes, we can proceed to configure the glassfish server and deploy the “brunch” web application to your localhost.

We put the brunch source code on the Desktop and refer “Desktop/brunch/code/glassfish4/glassfish” as the “as-install”. And we refer “Desktop/brunch/code/glassfish4” as the “as-install-parent”.

4.2.1 Start and Stop the Default Domain

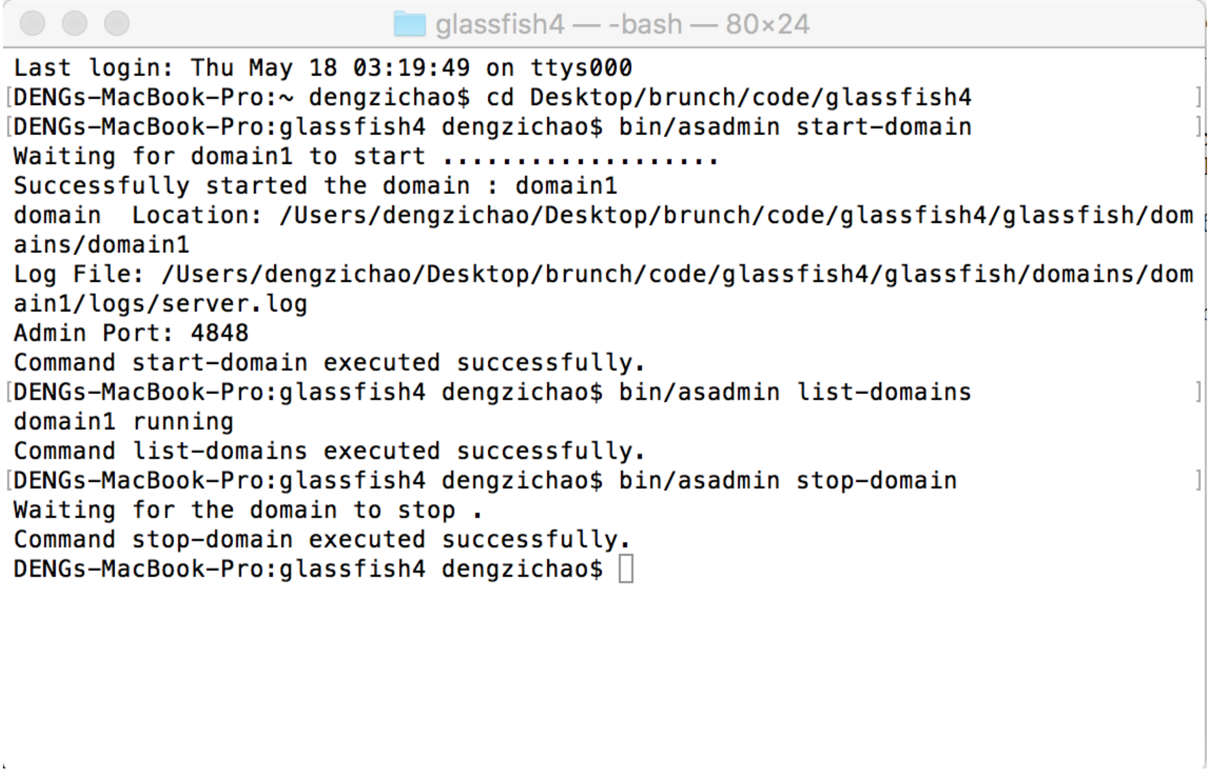
There is the default domain called domain1 automatically created when you download and configure the glassfish server. But the glassfish server software which is contained in the “brunch” source code package must be installed before you begin.

To start the default domain, run the asadmin start-domain command with the Macintosh terminal: as-install/bin/asadmin start-domain. This command will start the default domain, domain1 of glassfish.

To stop the default domain, issue the asadmin stop-domain command with the terminal: as-install/bin/asadmin stop-domain. Then your default domain will stop.

To check which domains are running on the glassfish server, issue the command: as-install/bin/asadmin list-domains

The Figure 37 shows the actual demo work with my Macintosh terminal.



```
Last login: Thu May 18 03:19:49 on ttys000
DENGs-MacBook-Pro:~ dengzichao$ cd Desktop/brunch/code/glassfish4
DENGs-MacBook-Pro:glassfish4 dengzichao$ bin/asadmin start-domain
Waiting for domain1 to start ...................
Successfully started the domain : domain1
domain  Location: /Users/dengzichao/Desktop/brunch/code/glassfish4/glassfish/dom
ains/domain1
Log File: /Users/dengzichao/Desktop/brunch/code/glassfish4/glassfish/domains/dom
ain1/logs/server.log
Admin Port: 4848
Command start-domain executed successfully.
DENGs-MacBook-Pro:glassfish4 dengzichao$ bin/asadmin list-domains
domain1 running
Command list-domains executed successfully.
DENGs-MacBook-Pro:glassfish4 dengzichao$ bin/asadmin stop-domain
Waiting for the domain to stop .
Command stop-domain executed successfully.
DENGs-MacBook-Pro:glassfish4 dengzichao$
```

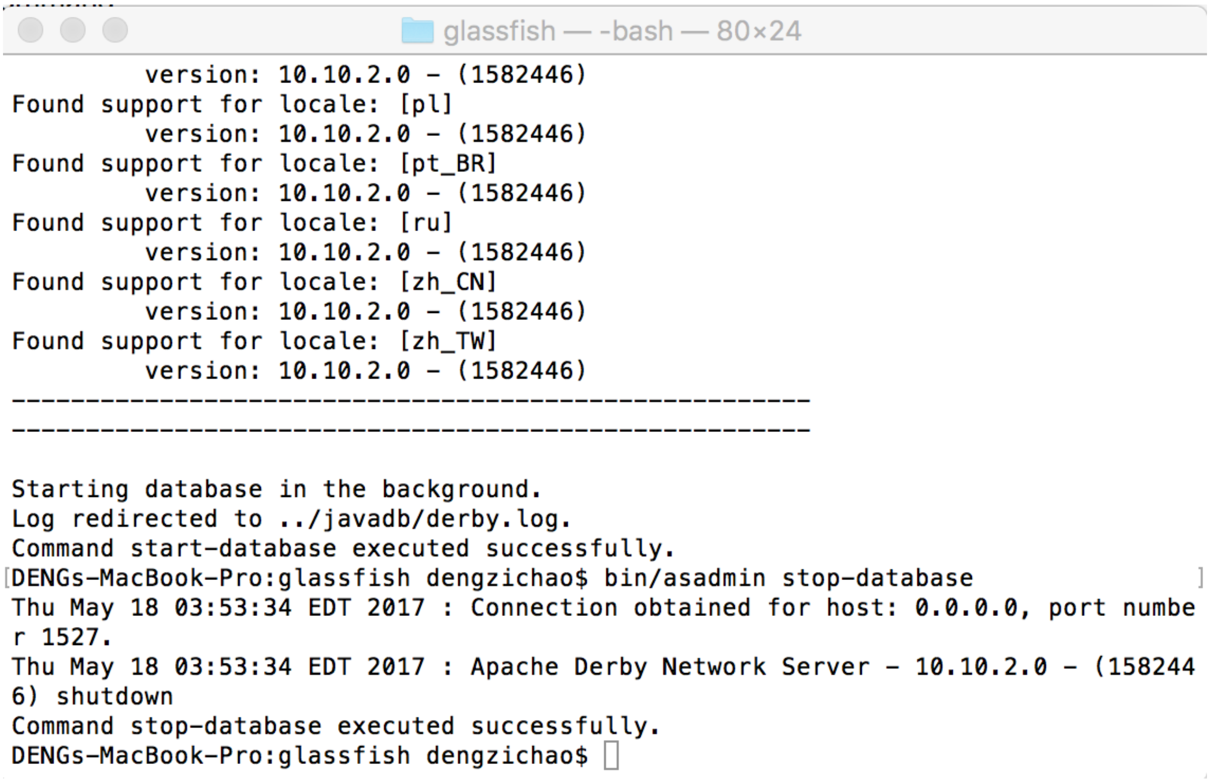
Figure 37. Start and stop the glassfish default domain

4.2.2 Start and Stop the Database Server

The database server will not run automatically when you start your glassfish server. If the web application requires a database backend, you need to start the database server manually. Before you start the database server, make sure that your default domain is running.

To start the Java DB server, issue the general command with your terminal:
as-install/bin/asadmin start-database --dbhome directory-path
For example, if we want to start the Java DB server from the default location, just issue: as-install/bin/asadmin start-database --dbhome as-install-parent/javadb.

To stop the Java DB server, issue the command: as-install/bin/asadmin stop-database. The example of a demo work is in Figure 38.



```
         version: 10.10.2.0 - (1582446)
Found support for locale: [pl]
         version: 10.10.2.0 - (1582446)
Found support for locale: [pt_BR]
         version: 10.10.2.0 - (1582446)
Found support for locale: [ru]
         version: 10.10.2.0 - (1582446)
Found support for locale: [zh_CN]
         version: 10.10.2.0 - (1582446)
Found support for locale: [zh_TW]
         version: 10.10.2.0 - (1582446)
------------------------------------------------------
------------------------------------------------------


Starting database in the background.
Log redirected to ../javadb/derby.log.
Command start-database executed successfully.
DENGs-MacBook-Pro:glassfish dengzichao$ bin/asadmin stop-database
Thu May 18 03:53:34 EDT 2017 : Connection obtained for host: 0.0.0.0, port numbe
r 1527.
Thu May 18 03:53:34 EDT 2017 : Apache Derby Network Server - 10.10.2.0 - (158244
6) shutdown
Command stop-database executed successfully.
DENGs-MacBook-Pro:glassfish dengzichao$ 
```

Figure 38. start and stop the java DB server

4.2.3 Start the Administration Console

The glassfish administration console offers the users a interface for configuring, managing and monitoring the glassfish server. Before you start your administration console, you must start the glassfish default domain. After that:

- Enter the URL for the administration console in the browser: http://localhost:4848 and the port 4848 is the administration console server port.
- If you decide to use a password when the glassfish is installed, you may need to log in to the administration console.

The Figure 39 shows the user interface of the administration console.

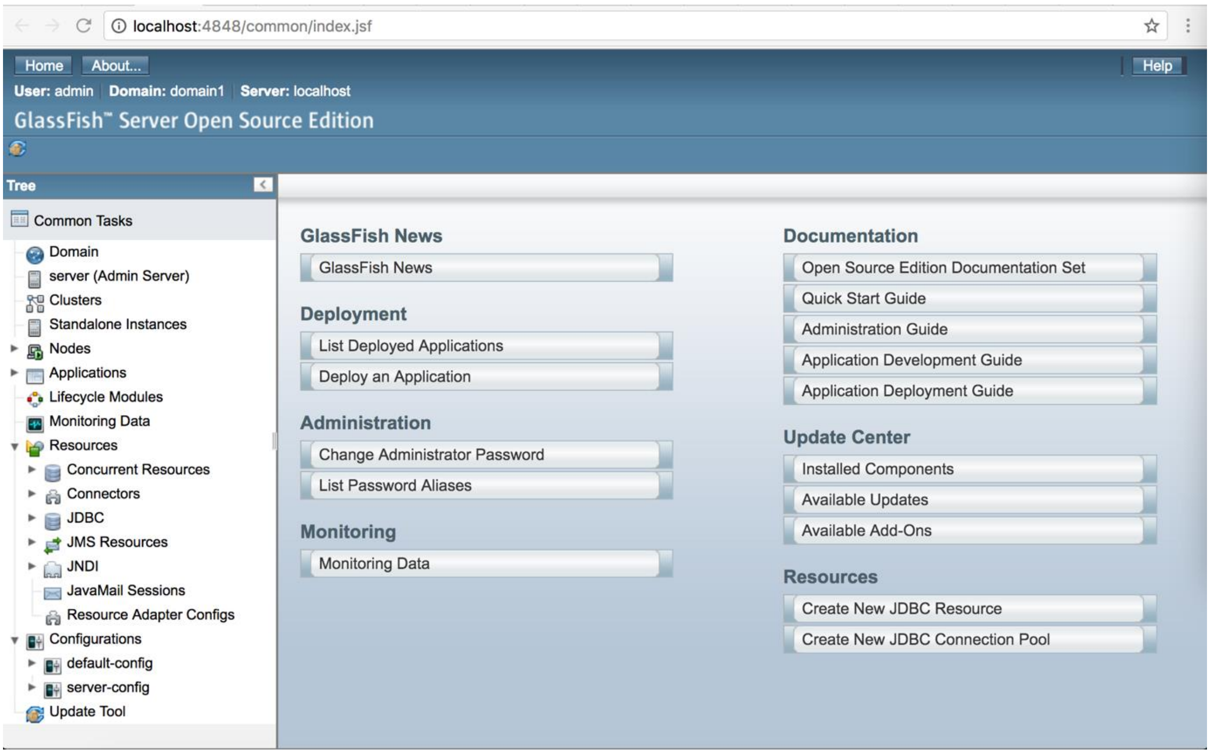


Figure 39. The administration console

4.2.4 Deploy and Undeploy the Web Application to GlassFish Server

Now we can proceed to deploy the “brunch” web application to the glassfish server and make it run on our localhost. First of all, we need to get the WEB.war contained in the brunch source codes which will be deployed to the glassfish server. And put the file WEB.war to the “as-install” directory. There are two ways to deploy web application to the glassfish server, via the command line or via the administration console.

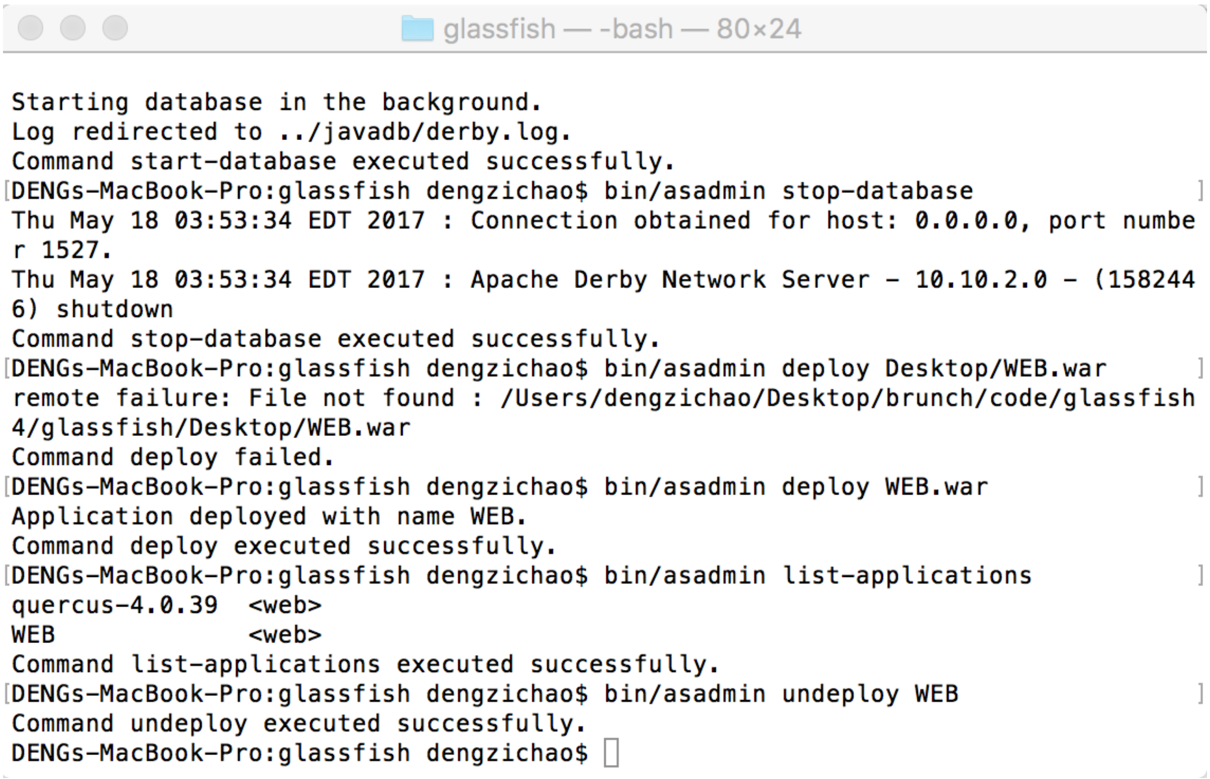
To deploy the web application via the command line:

- Issue the asadmin deploy command:
  as-install/bin/asadmin deploy war-name.
- Then we can enter the brunch web application via typing the following URL: http://localhost:8080/WEB. Port 8080 is for HTTP server port.

To list all the applications, issue the asadmin list-applications command: as-install/bin/asadmin list-applications;

To undeploy the brunch web application from the glassfish server, issue the asadmin undeploy command:
as-install/bin/asadmin undeploy war-name

The Figure 40 shows all the commands mentioned above.



```
Starting database in the background.
Log redirected to ../javadb/derby.log.
Command start-database executed successfully.
DENGs-MacBook-Pro:glassfish dengzichao$ bin/asadmin stop-database
Thu May 18 03:53:34 EDT 2017 : Connection obtained for host: 0.0.0.0, port numbe
r 1527.
Thu May 18 03:53:34 EDT 2017 : Apache Derby Network Server - 10.10.2.0 - (158244
6) shutdown
Command stop-database executed successfully.
DENGs-MacBook-Pro:glassfish dengzichao$ bin/asadmin deploy Desktop/WEB.war
remote failure: File not found : /Users/dengzichao/Desktop/brunch/code/glassfish
4/glassfish/Desktop/WEB.war
Command deploy failed.
DENGs-MacBook-Pro:glassfish dengzichao$ bin/asadmin deploy WEB.war
Application deployed with name WEB.
Command deploy executed successfully.
DENGs-MacBook-Pro:glassfish dengzichao$ bin/asadmin list-applications
quercus-4.0.39  <web>
WEB             <web>
Command list-applications executed successfully.
DENGs-MacBook-Pro:glassfish dengzichao$ bin/asadmin undeploy WEB
Command undeploy executed successfully.
DENGs-MacBook-Pro:glassfish dengzichao$ 
```

Figure 40. Deploy, Undeploy, List-applications command lines

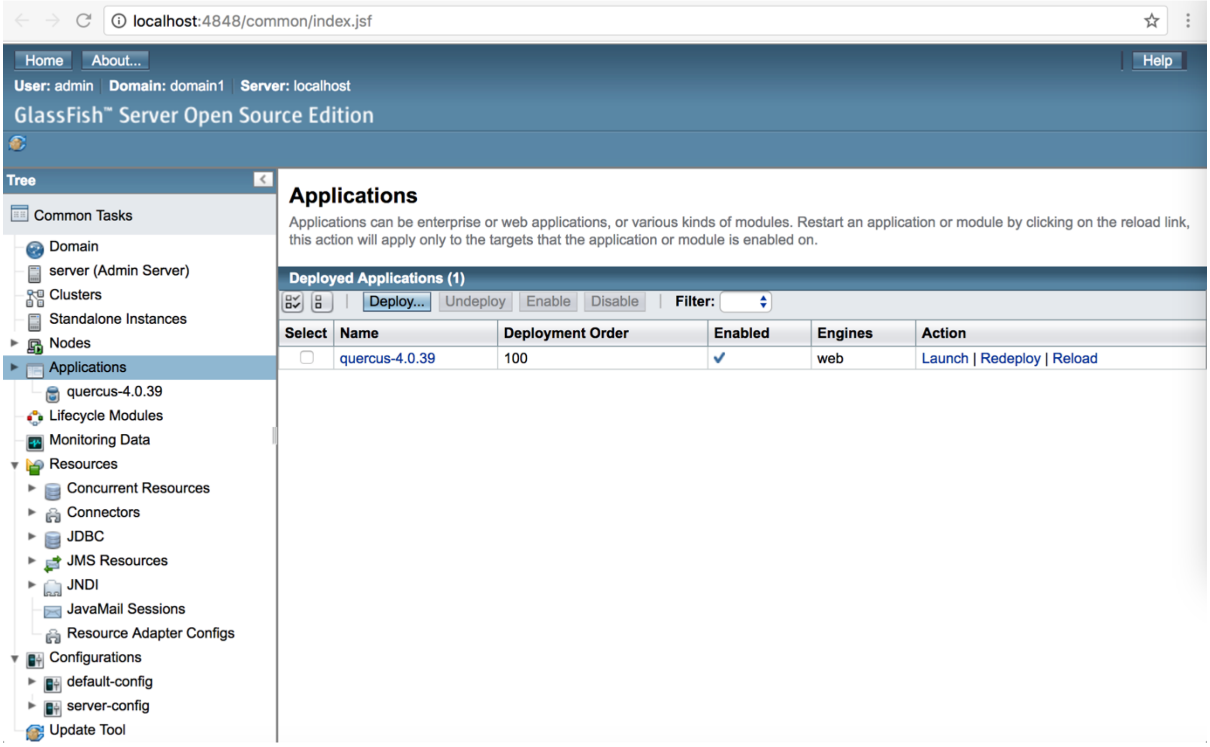
To deploy the brunch web application via the administration console:

- Access to the administration console by http://localhost:4848.
- Click the "Applications" node on the left column and lead you to the Application page, which is shown in Figure 41.
- Click the "Deploy" button and lead you to "Deploy Applications or Modules" page, which is shown in Figure 42.
- Click "Choose File" and select the WEB.war stored in the directory you select.
- Specify the Description filed to "WEB" and keep other settings the same.
- Click "OK" and now you can access to the brunch web application by http://localhost:8080/WEB.

To view the list of applications deployed to the glassfish server, just go to the administration console and click the "Applications" node on the left column. To Undeploy or Disable some web application, select the according web application and click the "Undeploy" or "Disable" button.

4.2.5 Deploy the Web Application Automatically

To deploy the brunch web application automatically, place the WEB.war to the directory of domain-dir/autodeploy, domain-dir refers to the domain you want to deploy your web application to. In our case, it is as-install/domains/domain1/autodeploy.



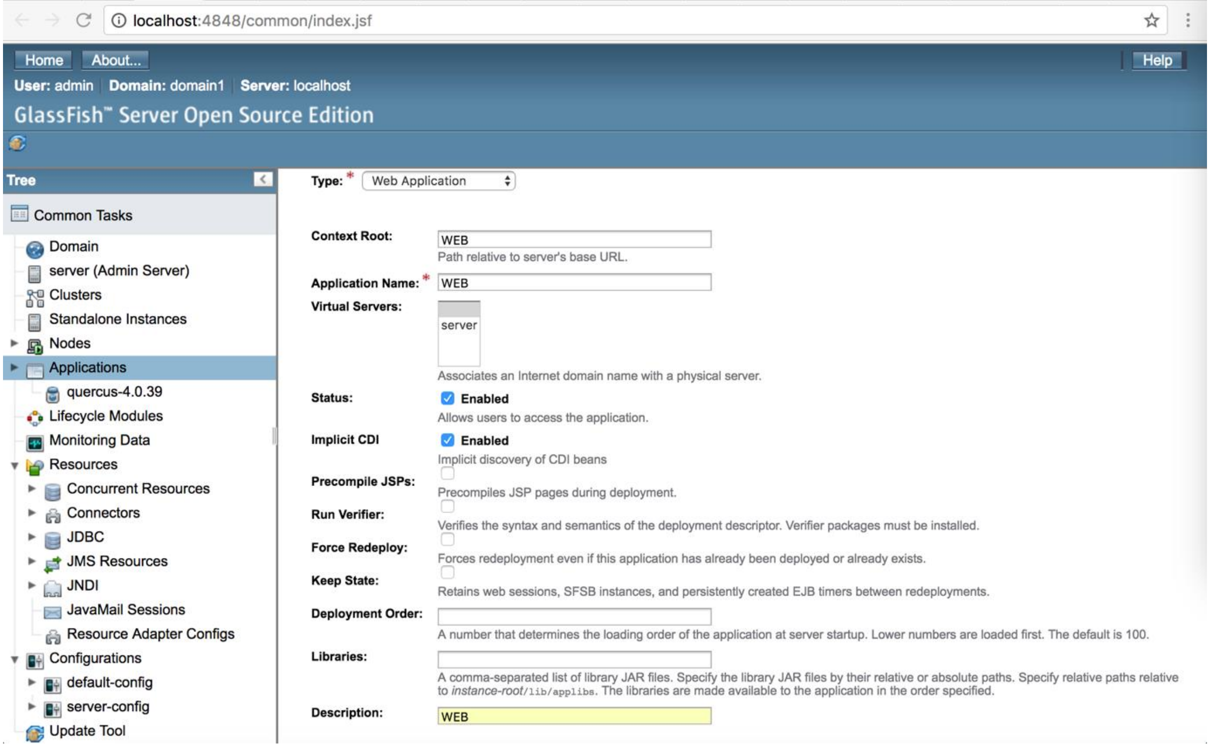
Applications

Applications can be enterprise or web applications, or various kinds of modules. Restart an application or module by clicking on the reload link, this action will apply only to the targets that the application or module is enabled on.

Deployed Applications (1)

Select	Name	Deployment Order	Enabled	Engines	Action
□	quercus-4.0.39	100	✓	web	Launch | Redeploy | Reload

Figure 41. Application page



Type: * Web Application

Context Root: WEB
Path relative to server's base URL.

Application Name: * WEB

Virtual Servers: server
Associates an Internet domain name with a physical server.

Status: ☑ Enabled
Allows users to access the application.

Implicit CDI ☑ Enabled
Implicit discovery of CDI beans

Precompile JSPs: □
Precompiles JSP pages during deployment.

Run Verifier: □
Verifies the syntax and semantics of the deployment descriptor. Verifier packages must be installed.

Force Redeploy: □
Forces redeployment even if this application has already been deployed or already exists.

Keep State: □
Retains web sessions, SFSB instances, and persistently created EJB timers between redeployments.

Deployment Order:
A number that determines the loading order of the application at server startup. Lower numbers are loaded first. The default is 100.

Libraries:
A comma-separated list of library JAR files. Specify the library JAR files by their relative or absolute paths. Specify relative paths relative to *instance-root*/lib/applibs. The libraries are made available to the application in the order specified.

Description: WEB

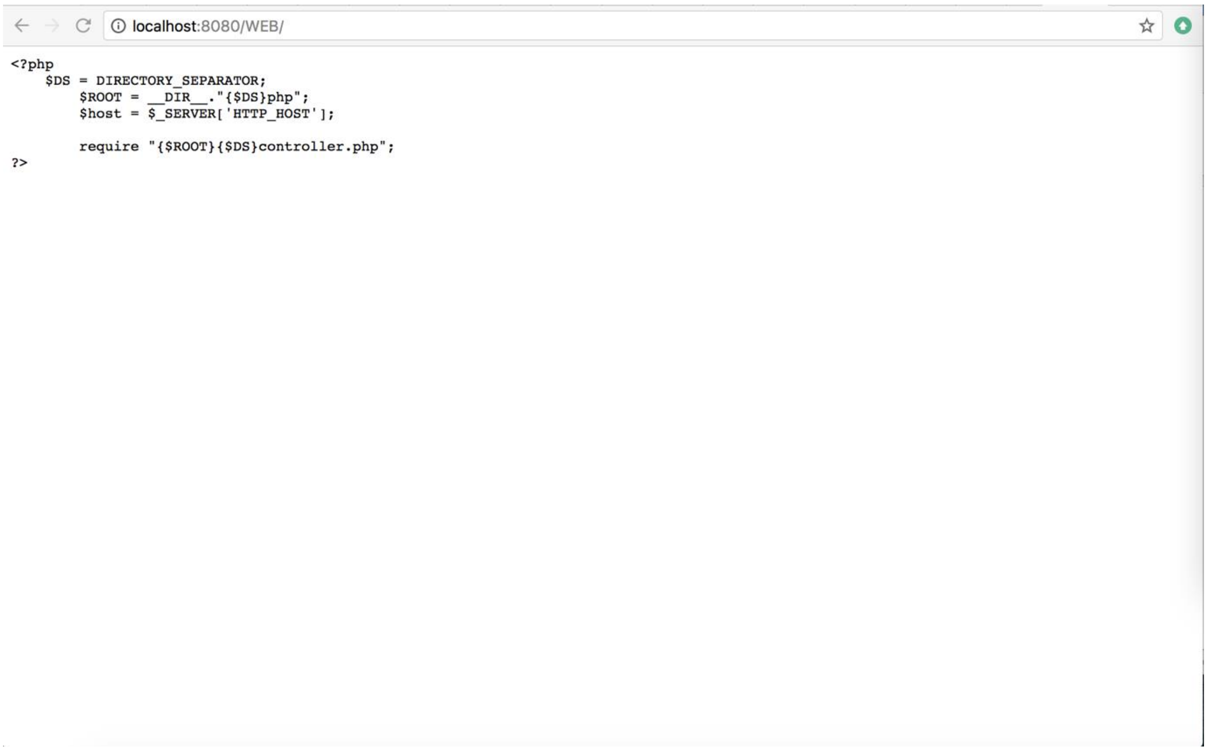
Figure 42. “Deploy Applications or Modules” page

Chapter 5. Debug, Solving Exceptions and Improve “brunch” Web Application

After deploying the “brunch” web application to the glassfish server, there are still some remaining bugs, exceptions in the localhost. This chapter is to mainly talk about how I debug and deal with different exceptions to improve the brunch web application. I primarily focus on the web application programming and database programming.

5.1 Quercus PHP Interpreter

When I deployed the brunch web application to the glassfish at the beginning, the glassfish server always returned me the page like this, shown in Figure 43 after I entered the http://localhost:8080/WEB.



```
<?php
    $DS = DIRECTORY_SEPARATOR;
        $ROOT = __DIR__."{$DS}php";
        $host = $_SERVER['HTTP_HOST'];

        require "{$ROOT}{$DS}controller.php";
?>
```

Figure 43. The error page of brunch web application

This exception happens because the PHP code of brunch web application cannot be interpreted. Therefore, the Quercus PHP interpreter should be used to display the PHP codes.

Quercus is Caucho Technology’s 100% Java implementation of PHP 5 released under the Open Source GPL license. Quercus comes with many PHP modules and extensions like PDF, PDO, MySQL, and JSON. With Quercus, PHP applications automatically take advantage of Java application server features just as connection pooling and clustered sessions.

Quercus presents a new mixed Java/PHP approach to web applications and services where Java and PHP tightly integrate with each other. PHP applications can choose to use Java libraries and technologies like JMS, EJB, SOA frameworks, Hibernate, and Spring. This revolutionary capability is made possible because:

- PHP code is interpreted/compiled into Java.
- Quercus and its libraries are written entirely in Java.

This architecture allows PHP applications and Java libraries to talk directly with one another at the program level. To facilitate this new Java/PHP architecture, Quercus provides and API and interface to expose Java libraries to PHP. [6]

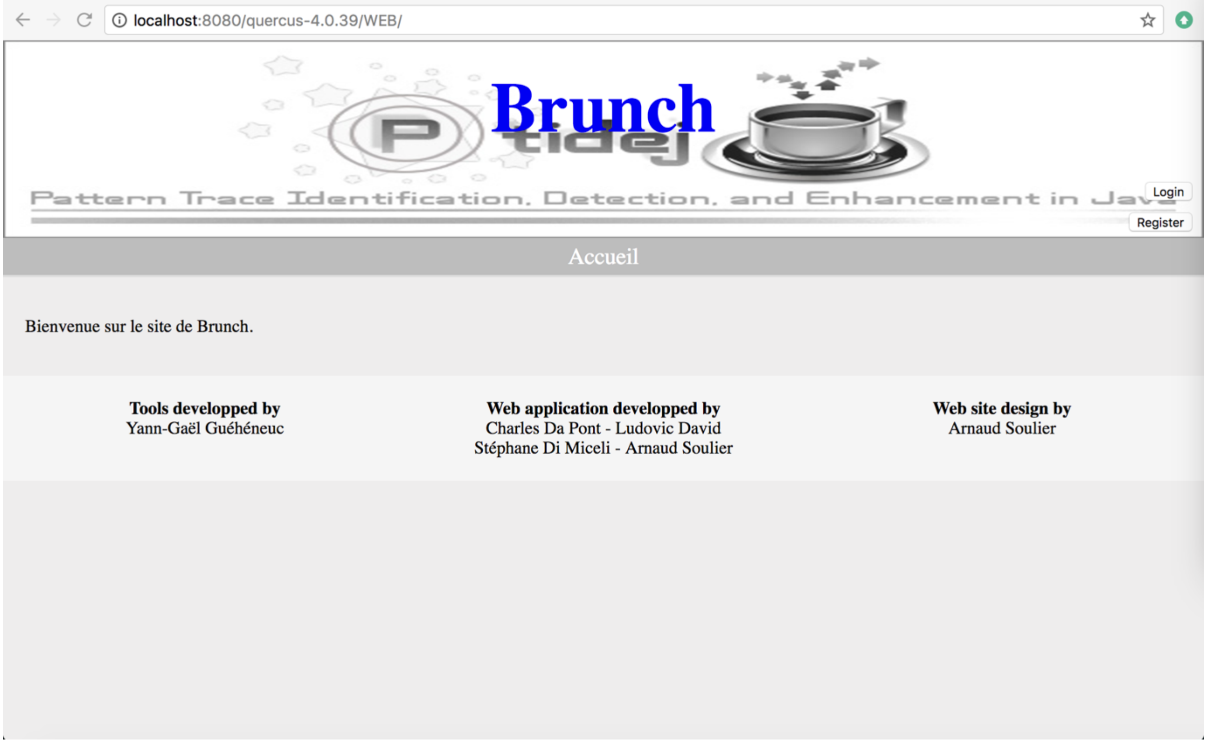
First of all, we need to deploy the Quercus PHP interpreter to the glassfish server.

- Obtain the Quercus PHP interpreter from http://quercus.caucho.com/.
- Obtain the file quercus-4.0.39.war and deploy the war file to the glassfish server. Follow the chapter 4.2 to do the deployment.
- In order to confirm the PHP interpreter works well, issue the URL:

  http://localhost:8080/quercus-4.0.39/,

  Then you can view the default PHP script of Quercus PHP interpreter.
- In order to make the brunch web application interpreted, transfer the brunch PHP application to the subdirectory of the Quercus directory. Now the Quercus's directory is: as-install/domains/doamin1/application/. And just put the entire WEB PHP application folder to the quercus-4.0.39 under the applications.
- In the end, enter the URL: http://localhost:8080/ quercus-4.0.39/WEB/ and you can check whether the brunch PHP application can display or not.

The Figure 44 shows the brunch web application interface after installing Quercus successfully.


localhost:8080/quercus-4.0.39/WEB/
Brunch
Ptidej
Pattern Trace Identification, Detection, and Enhancement in Java
Login
Register
Accueil

Bienvenue sur le site de Brunch.

Tools developped by
Yann-Gaël Guéhéneuc

Web application developped by
Charles Da Pont - Ludovic David
Stéphane Di Miceli - Arnaud Soulier

Web site design by
Arnaud Soulier

Figure 44. brunch web application interface

5.2 Integrate the JDBC Driver

After the interface of "brunch" web application successfully displayed via the glassfish server, there are still remaining exceptions. The first exception I got is called "java.lang.ClassNotFoundException: com.mysql.jdbc.Driver". We can short it to “com.mysql.jdbc.Driver”. After some researches about this exception, I gradually learnt how to deal with this exception.

First of all, the JDBC is the short of Java Database Connectivity. JDBC is an application programming interface (API) for the Java programming language. JDBC defines how the user can get into the database. Because there is a database called “brunch” in this web application we work on. Therefore, JDBC is essential and crucial.

There is also a software component called JDBC driver required, which makes the Java application can communicate and interact with the database. In order to interact with database, the JDBC needs the JDBC driver, which can give the connection from the Java application to each database and provides some rules or protocols to transmit query and response from the client or application to the database.

Therefore, the JDBC driver is crucial and you need to choose JDBC driver to work with the glassfish server, which means that you must configure the JDBC driver by yourself:

- You can choose the JDBC drivers which are tested fully and supported by Oracle.
- Download the JDBC driver from the JDBC/UCP Download Page of Oracle:
  http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html.
- Make the JDBC driver .jar file called mysql-connector-java-5.0.8-bin.jar accessible by putting it in the library of glassfish directory, in our case, it is as-install/lib. As a result, all the applications or modules which are deployed on the glassfish server can make use of the JDBC driver and get access to the databases.

This exception is solved successfully.

One thing needs to be noted is that the glassfish server administration console can automatically detect the JDBC driver when you create a JDBC connection pool, the methods to create a JDBC connection pool will be covered later.

There are some other guesses of this exception:

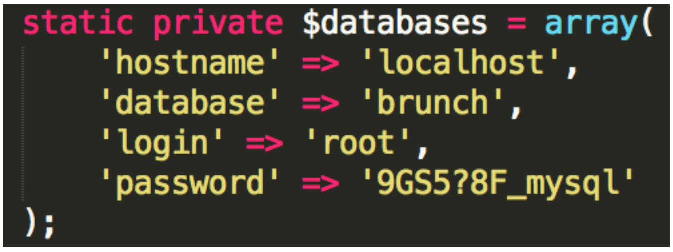
- The JDBC driver, mysql-connector-java-5.0.8-bin.jar has already been installed and integrated in the class path, however, the class path is overridden.
- The JDBC driver, mysql-connector-java-5.0.8-bin.jar is already in your class path but this class path does not have the read permission.

5.3 Set up MySQL database and JDBC Connection Pool for brunch Web Application

Here comes another exception called “java.sql.SQLException: Access denied for user ‘root’ @ ‘localhost’ (using password: YES)”, which can be short for “Access denied for user ‘root’ @ ‘localhost’”. This exception is mainly caused because that the brunch web application cannot get access to the MySQL server, which may require you to set up a database for brunch web application and give it the authorization to access to this MySQL database.

SQL is the short of structure Query Language, which a high-level language. It is designed to interact with the relational databases. There are a lot of commands in the SQL including SELECT, INSERT, UPDATE, CREATE TABLE, DROP TABLE and etc.

A relational database system contains a lot of databases. A database contains a lot of tables. And a table comprises rows and columns. The information regarding setting up the MySQL server can be found in the Conf.php of brunch source code shown in Figure 45.



```
static private $databases = array(
    'hostname' => 'localhost',
    'database' => 'brunch',
    'login' => 'root',
    'password' => '9GS5?8F_mysql'
);
```

Figure 45. Conf.php

Now we can download and install the MySQL server:

- Download and install the MySQL from https://dev.mysql.com/downloads/mysql/.
- Start/Shut down the MySQL server: System Preferences --> MySQL --> Start or Stop.
- Start a "client": issue the commands: cd /user/local/mysql/bin
  ./mysql -u root -p
  Enter password: (can be found in Conf.php).

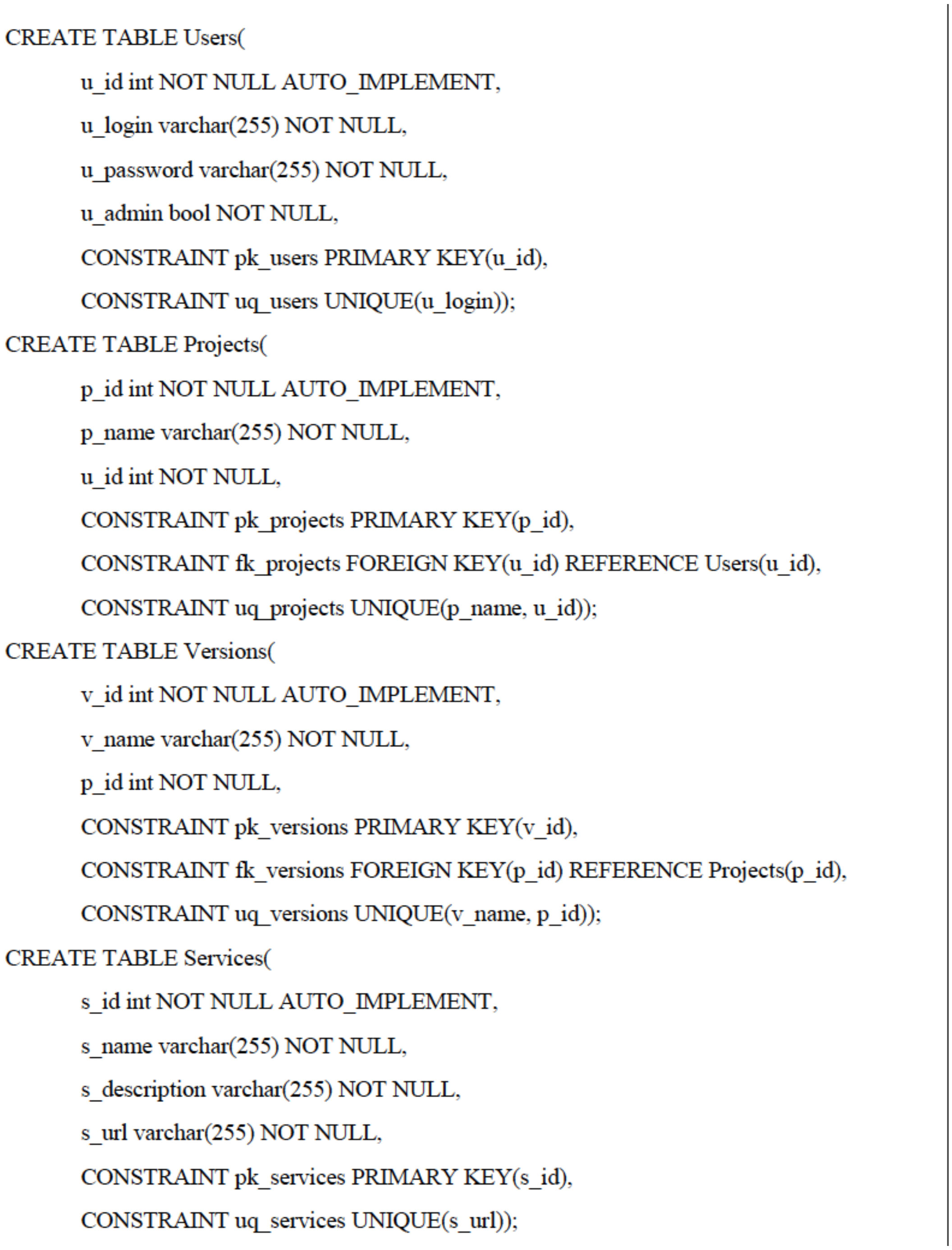
Then we can proceed to set up the brunch database:

- Create the database called brunch: create database if not exists brunch;
- List all the databases in the server: show databases;
- Use the "brunch" database as the default database, you can refer to the table in the default database by just using the table name: use brunch;

Now, the brunch database is already set up and changed to the default database.

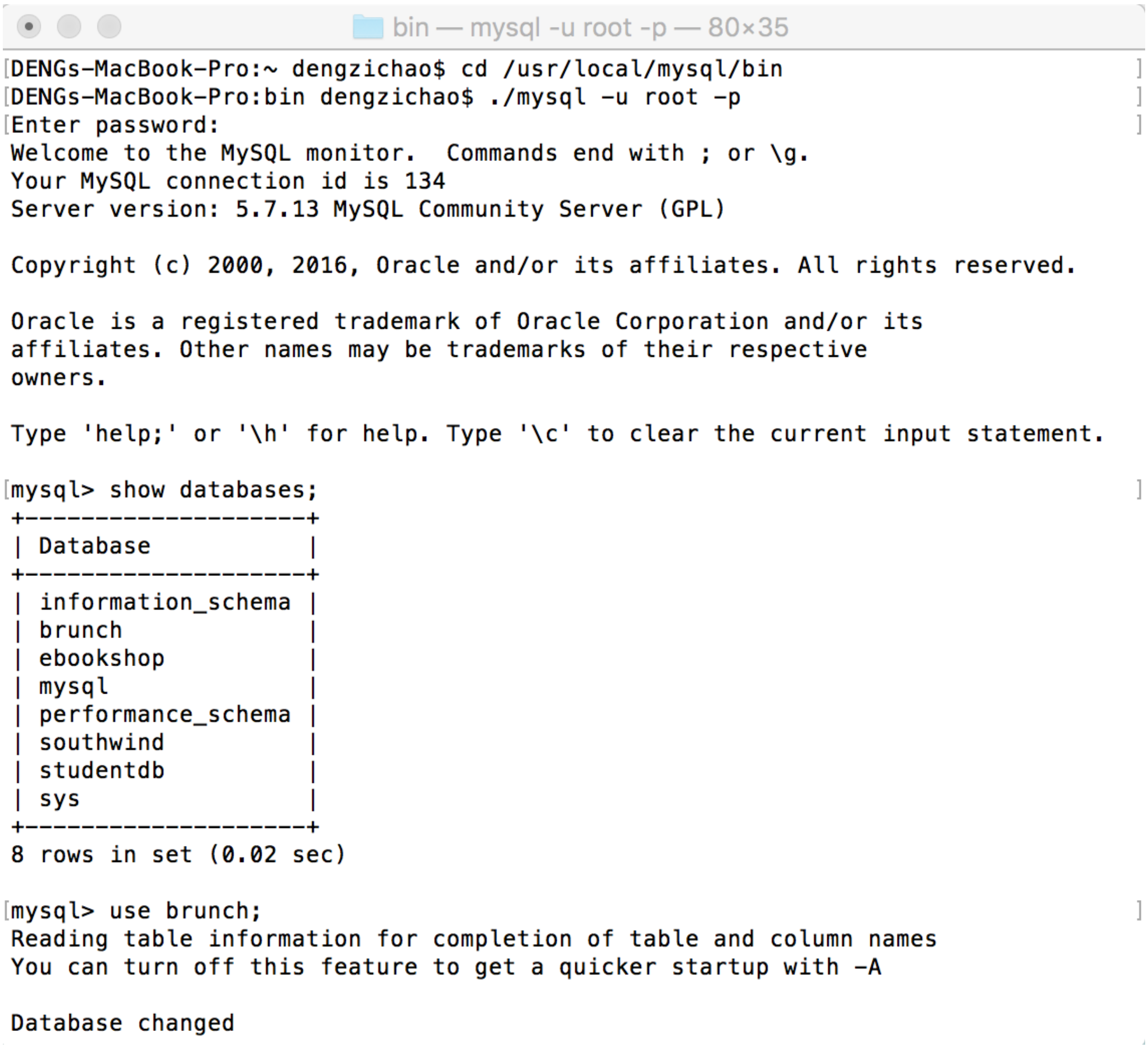
The brunch database is used by the brunch web application to store the data of created users, projects, versions and services information. Therefore, we need to set up 4 tables named Users, Projects, Versions and Services and specify the rows and columns inside each table.

The detailed statements are shown in Figure 46.



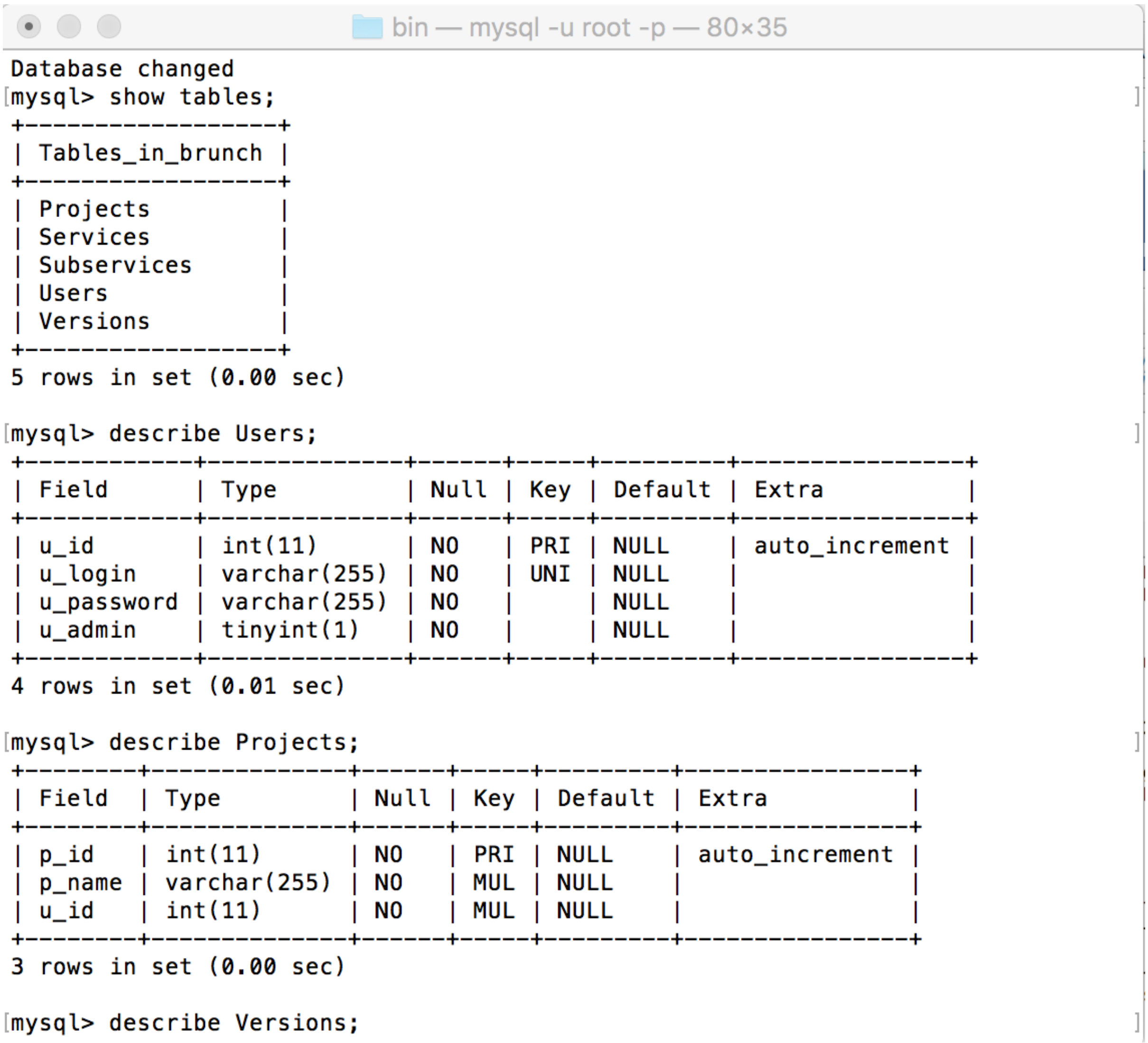
```
CREATE TABLE Users(
        u_id int NOT NULL AUTO_IMPLEMENT,
        u_login varchar(255) NOT NULL,
        u_password varchar(255) NOT NULL,
        u_admin bool NOT NULL,
        CONSTRAINT pk_users PRIMARY KEY(u_id),
        CONSTRAINT uq_users UNIQUE(u_login));
CREATE TABLE Projects(
        p_id int NOT NULL AUTO_IMPLEMENT,
        p_name varchar(255) NOT NULL,
        u_id int NOT NULL,
        CONSTRAINT pk_projects PRIMARY KEY(p_id),
        CONSTRAINT fk_projects FOREIGN KEY(u_id) REFERENCE Users(u_id),
        CONSTRAINT uq_projects UNIQUE(p_name, u_id));
CREATE TABLE Versions(
        v_id int NOT NULL AUTO_IMPLEMENT,
        v_name varchar(255) NOT NULL,
        p_id int NOT NULL,
        CONSTRAINT pk_versions PRIMARY KEY(v_id),
        CONSTRAINT fk_versions FOREIGN KEY(p_id) REFERENCE Projects(p_id),
        CONSTRAINT uq_versions UNIQUE(v_name, p_id));
CREATE TABLE Services(
        s_id int NOT NULL AUTO_IMPLEMENT,
        s_name varchar(255) NOT NULL,
        s_description varchar(255) NOT NULL,
        s_url varchar(255) NOT NULL,
        CONSTRAINT pk_services PRIMARY KEY(s_id),
        CONSTRAINT uq_services UNIQUE(s_url));
```

Figure 46. Create table statements



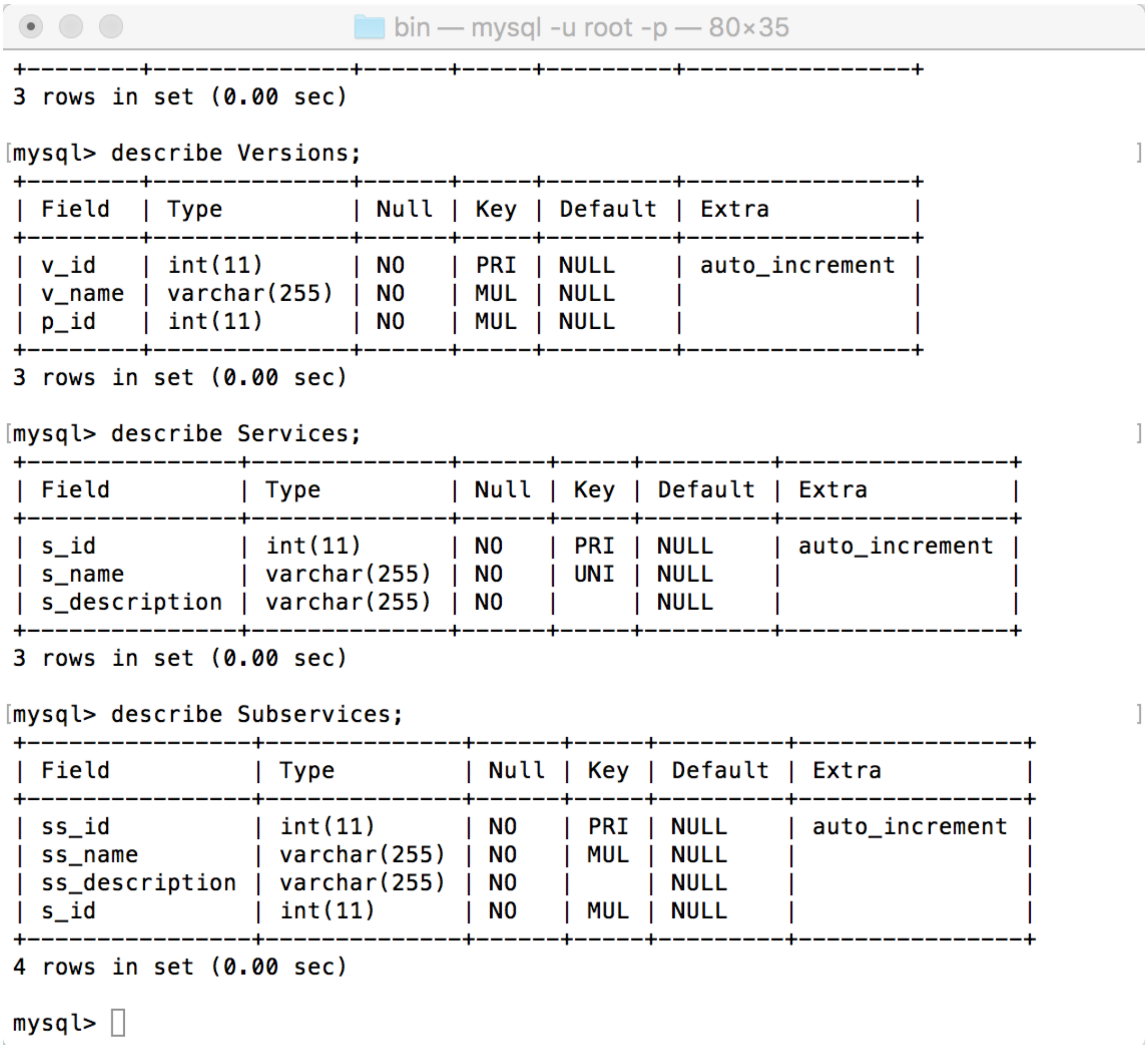
```
DENGs-MacBook-Pro:~ dengzichao$ cd /usr/local/mysql/bin
DENGs-MacBook-Pro:bin dengzichao$ ./mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 134
Server version: 5.7.13 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| brunch             |
| ebookshop          |
| mysql              |
| performance_schema |
| southwind          |
| studentdb          |
| sys                |
+--------------------+
8 rows in set (0.02 sec)

mysql> use brunch;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

Figure 47. show databases



```
Database changed
mysql> show tables;
+-------------------+
| Tables_in_brunch |
+-------------------+
| Projects          |
| Services          |
| Subservices       |
| Users             |
| Versions          |
+-------------------+
5 rows in set (0.00 sec)

mysql> describe Users;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| u_id       | int(11)      | NO   | PRI | NULL    | auto_increment |
| u_login    | varchar(255) | NO   | UNI | NULL    |                |
| u_password | varchar(255) | NO   |     | NULL    |                |
| u_admin    | tinyint(1)   | NO   |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)

mysql> describe Projects;
+--------+--------------+------+-----+---------+----------------+
| Field  | Type         | Null | Key | Default | Extra          |
+--------+--------------+------+-----+---------+----------------+
| p_id   | int(11)      | NO   | PRI | NULL    | auto_increment |
| p_name | varchar(255) | NO   | MUL | NULL    |                |
| u_id   | int(11)      | NO   | MUL | NULL    |                |
+--------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> describe Versions;
```

Figure 48. show tables (1)



```
+--------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

[mysql> describe Versions;
+--------+--------------+------+-----+---------+----------------+
| Field  | Type         | Null | Key | Default | Extra          |
+--------+--------------+------+-----+---------+----------------+
| v_id   | int(11)      | NO   | PRI | NULL    | auto_increment |
| v_name | varchar(255) | NO   | MUL | NULL    |                |
| p_id   | int(11)      | NO   | MUL | NULL    |                |
+--------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

[mysql> describe Services;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| s_id          | int(11)      | NO   | PRI | NULL    | auto_increment |
| s_name        | varchar(255) | NO   | UNI | NULL    |                |
| s_description | varchar(255) | NO   |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

[mysql> describe Subservices;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| ss_id          | int(11)      | NO   | PRI | NULL    | auto_increment |
| ss_name        | varchar(255) | NO   | MUL | NULL    |                |
| ss_description | varchar(255) | NO   |     | NULL    |                |
| s_id           | int(11)      | NO   | MUL | NULL    |                |
+----------------+--------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)

mysql> 
```

Figure 49. show tables (2)

Figure 47, 48, 49 show the real work on the terminal: show databases; show tables in brunch; describe tables of Users, Projects, Versions, Services, Subservices.

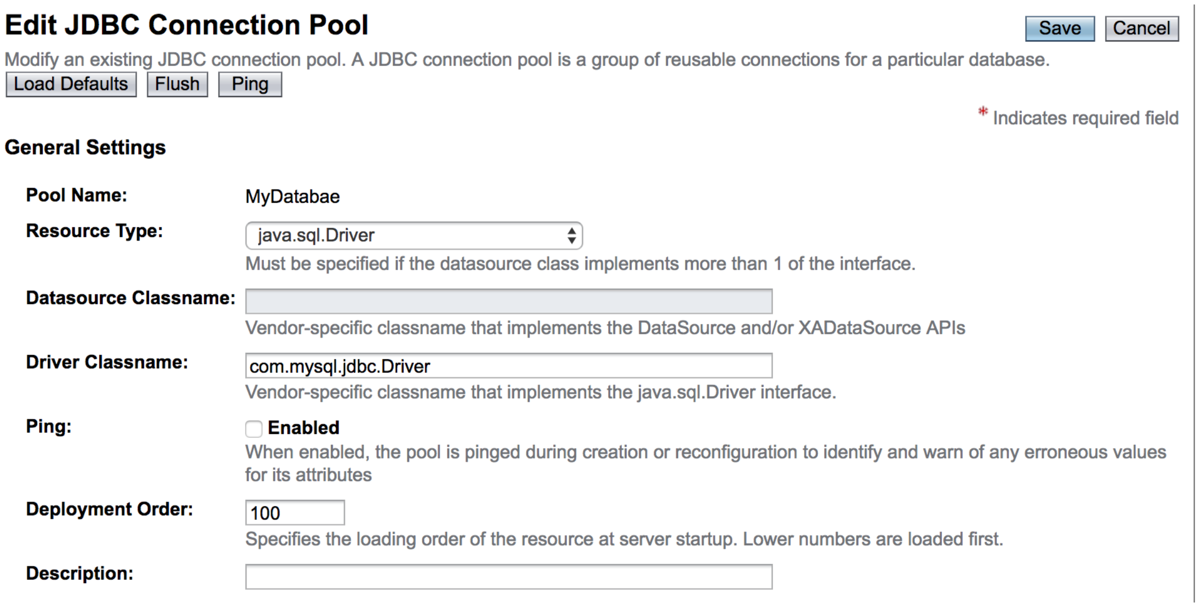
Now the brunch database has already been configured. In order to use this database, a JDBC connection pool is also required to set up.

A connection is created whenever the web application needs to interact with the database. But it is troublesome to create connection each time. To facilitate the reuse of connection, a memory cache storing the connections, which is called JDBC connection pool is maintained by the connection pool module as a layer on top of any JDBC driver product.

In order to configure a JDBC connection pool:

- Open the glassfish server administration console, in the left navigation column, find "Resources -- JDBC – JDBC Connection Pools", click New to set up a new JDBC connection pool.

- In the general settings, remember to specify the pool name, choose the java.sql.Driver as the resource type, and remain other settings unchanged. Click Next.
- Specify the Driver Classname, which is the Driver specific. Consult the driver you chose to select the right class. Here, in our case, we choose com.mysql.jdbc.Driver. Click Finish to create a JDBC connection pool. The user interface is shown in Figure 50.
- You can edit the additional properties of the JDBC connection pool including the password, user and URL. In our case, password: 9GS5?8F_mysql; user: root and URL: jdbc:mysql://localhost/brunch?user=root&password=9GS5?8F_mysql. The interface is shown in Figure 51.
- At last, create the JDBC Resources that you can refer in the web application by clicking the JDBC – JDBC Resources. Specify the JNDI Name as well as the Pool Name. The interface is shown in Figure 50.



Modify an existing JDBC connection pool. A JDBC connection pool is a group of reusable connections for a particular database.

* Indicates required field

General Settings

Pool Name: MyDatabae

Resource Type: java.sql.Driver
Must be specified if the datasource class implements more than 1 of the interface.

Datasource Classname:
Vendor-specific classname that implements the DataSource and/or XADataSource APIs

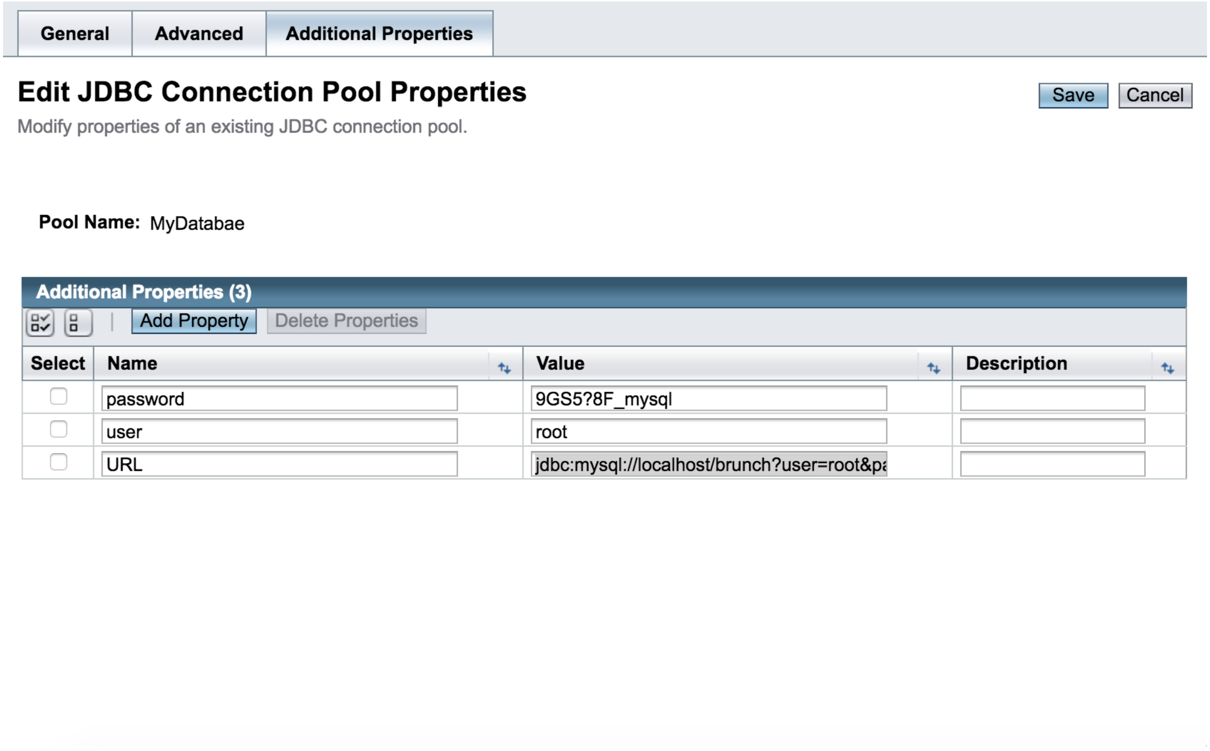
Driver Classname: com.mysql.jdbc.Driver
Vendor-specific classname that implements the java.sql.Driver interface.

Ping: □ Enabled
When enabled, the pool is pinged during creation or reconfiguration to identify and warn of any erroneous values for its attributes

Deployment Order: 100
Specifies the loading order of the resource at server startup. Lower numbers are loaded first.

Description:

Figure 50. Create JDBC connection pool



Edit JDBC Connection Pool Properties

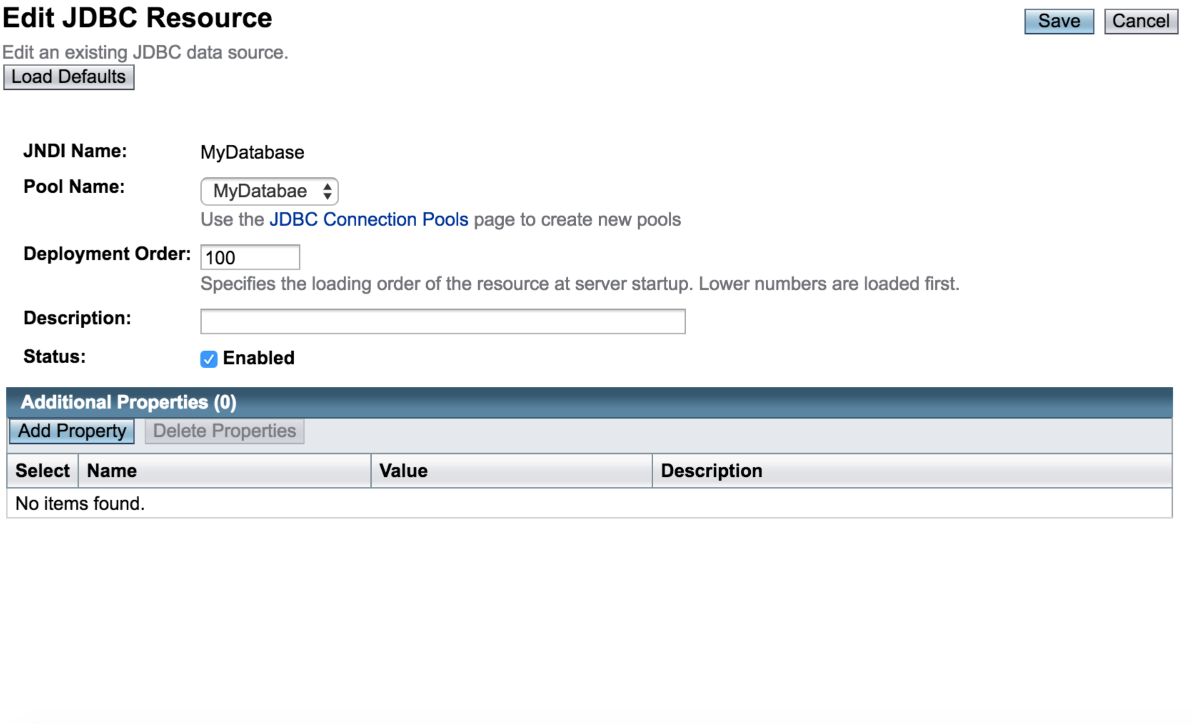
Modify properties of an existing JDBC connection pool.

Pool Name: MyDatabae

Additional Properties (3)

Select	Name	Value	Description
□	password	9GS5?8F_mysql	
□	user	root	
□	URL	jdbc:mysql://localhost/brunch?user=root&pa	

Figure 51. Edit the JDBC connection pool



Edit JDBC Resource

Edit an existing JDBC data source.

JNDI Name: MyDatabase

Pool Name: MyDatabae

Use the JDBC Connection Pools page to create new pools

Deployment Order: 100

Specifies the loading order of the resource at server startup. Lower numbers are loaded first.

Description:

Status: ☑ Enabled

Additional Properties (0)

Select	Name	Value	Description
No items found.			

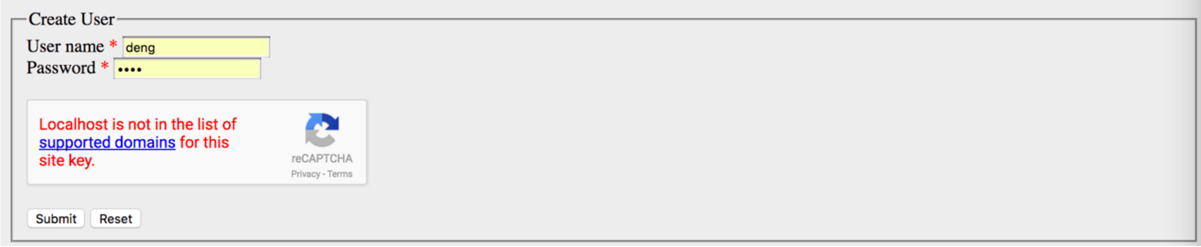
Figure 52. Edit JDBC Resource

Finally, we have created a JDBC connection pool and can use it in the web application. The exception has been solved successfully.

5.4 Google reCAPTCHA

reCAPTCHA is a free service that protects your website from spam and abuse. reCAPTCHA uses an advanced risk analysis engine and adaptive CAPTCHAs to keep automated software from engaging in abusive activities on your site. It does this while letting your valid users pass through with ease.

reCAPTCHA offers more than just spam protection. Every time our CAPTCHAs are solved, that human effort help digitize text, annotate images, and build machine learning datasets. This in turn helps preserve books, improve maps, and solve hard AI problems. [7] Therefore, the brunch web application uses google reCAPTCHA to provide protection from attacks, abuse and spam as shown in Figure 53.



Create User

User name * deng

Password * ••••

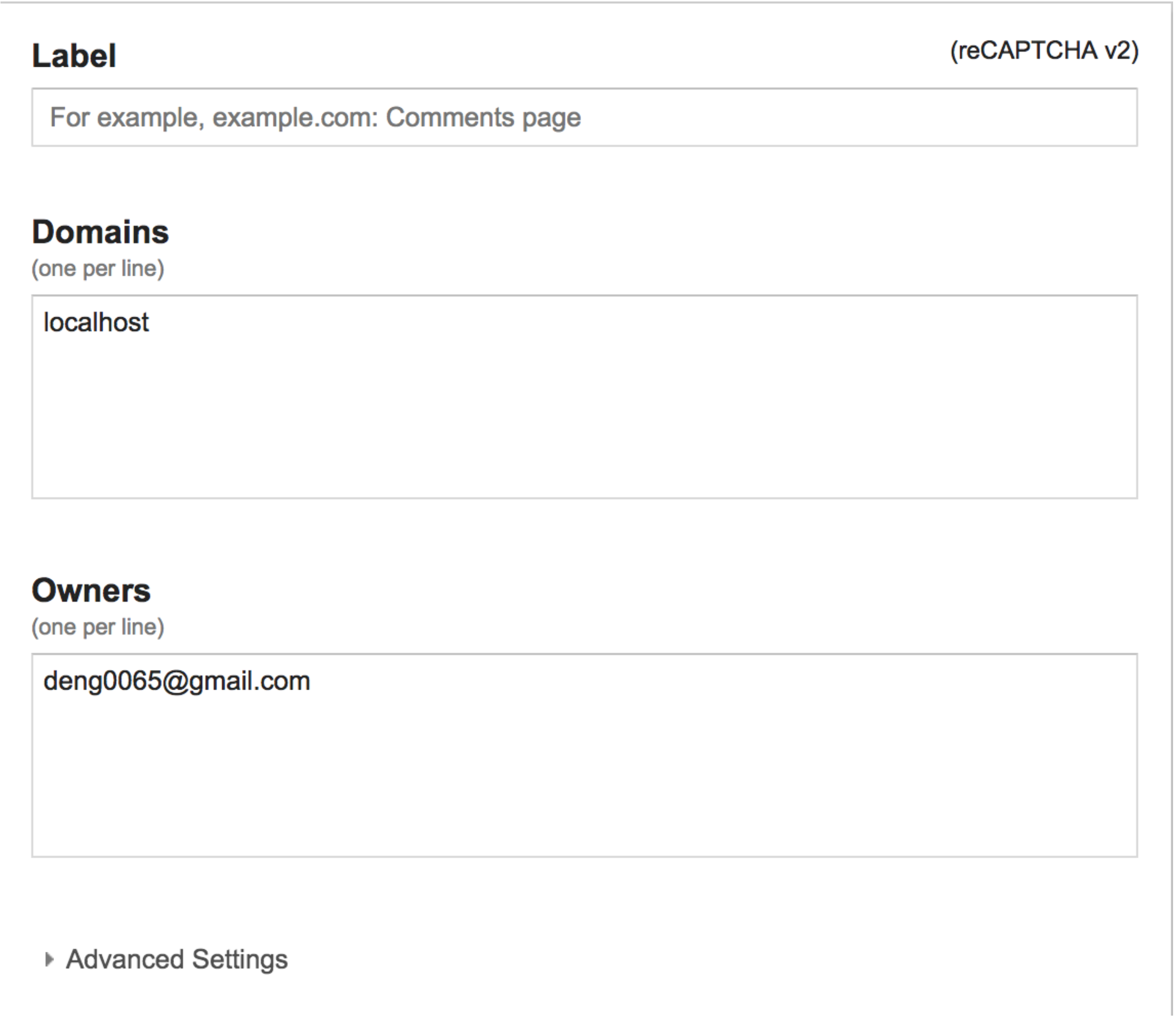
Localhost is not in the list of supported domains for this site key.

Figure 53. google reCAPTCHA

However, here is an exception that: localhost is not in the list of supported domains for this site key. Generally, there is a pair of API key for using google reCAPTCHA, the site key and the secret key. The site key is used to display the widgets and user interface on the website. The secret key is used to interact between the web application and the reCAPTCHA server to authenticate the user's response. Therefore, this secret needs to be kept safe in order to make sure the security.

Now the localhost is not in the supported domain of the site key. We need to register a new API key pair for the domain localhost. Here is the procedure:

- Go to the sign up page: https://www.google.com/recaptcha/admin#list.
- Choose the type of reCAPTCHA of reCAPTCHA V2.
- Type in "localhost" in the field of Domains. Each API key pair is unique to the domains and first-level subdomains that you specify. Therefore, if you want that "localhost" in the supported domain, you are required to add the "localhost" to the list of domains. The Figure 54 shows the interface of signing up the API key pair.
- Then a pair of API key pair unique to your domains will be created for you as shown in Figure 55.



Label (reCAPTCHA v2)

For example, example.com: Comments page

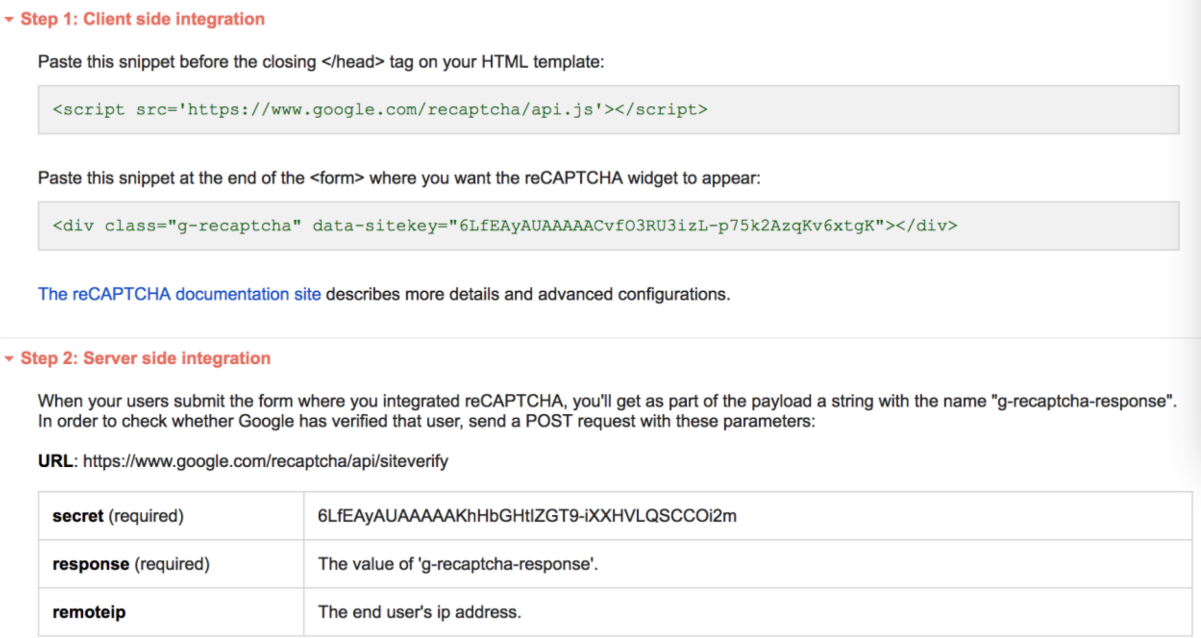
Domains
(one per line)

localhost

Owners
(one per line)

deng0065@gmail.com

Figure 54. Sign up the API key pair



▾ Step 1: Client side integration

Paste this snippet before the closing </head> tag on your HTML template:

```
<script src='https://www.google.com/recaptcha/api.js'></script>
```

Paste this snippet at the end of the <form> where you want the reCAPTCHA widget to appear:

```
<div class="g-recaptcha" data-sitekey="6LfEAyAUAAAAACvfO3RU3izL-p75k2AzqKv6xtgK"></div>
```

The reCAPTCHA documentation site describes more details and advanced configurations.

▾ Step 2: Server side integration

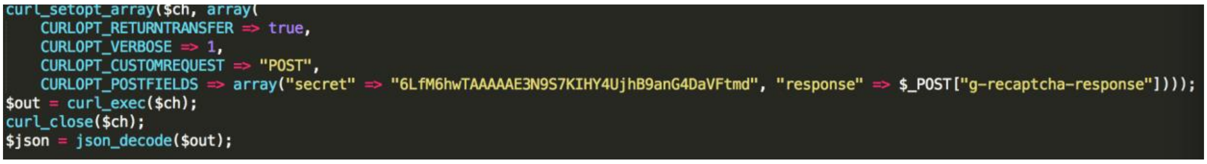
When your users submit the form where you integrated reCAPTCHA, you'll get as part of the payload a string with the name "g-recaptcha-response". In order to check whether Google has verified that user, send a POST request with these parameters:

URL: https://www.google.com/recaptcha/api/siteverify

secret (required)	6LfEAyAUAAAAAKhHbGHtlZGT9-iXXHVLQSCCOi2m
response (required)	The value of 'g-recaptcha-response'.
remoteip	The end user's ip address.

Figure 55. API key pair

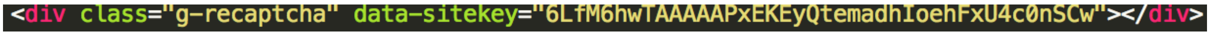
Now you need to replace this alternative key pairs in the PHP codes. In viewCreatedUser.php, replace the key "6LfM6hwTAAAAAE3N9S7KIHY4UjhB9anG4DaVFtmd” by the new secret key “6LfEAyAUAAAAAKhHbGHtIZGT9-iXXHVLQSCCOi2m”.



```
curl_setopt_array($ch, array(
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_VERBOSE => 1,
    CURLOPT_CUSTOMREQUEST => "POST",
    CURLOPT_POSTFIELDS => array("secret" => "6LfM6hwTAAAAAE3N9S7KIHY4UjhB9anG4DaVFtmd", "response" => $_POST["g-recaptcha-response"])));
$out = curl_exec($ch);
curl_close($ch);
$json = json_decode($out);
```

Figure 56. viewCreatedUser.php

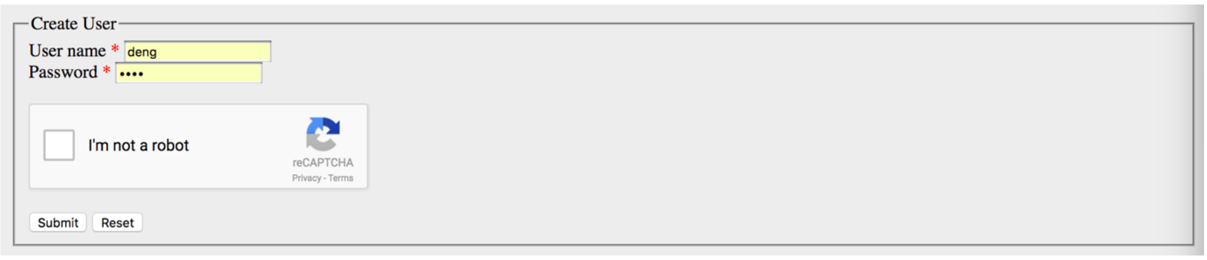
And then, replace the site key "6LfM6hwTAAAAAPxEKEyQtemadhIoehFxU4c0nSCw" by the new site key “6LfEAyAUAAAAACvfO3RU3izL-p75k2AzqKv6xtgK” in the following piece of codes.



```
<div class="g-recaptcha" data-sitekey="6LfM6hwTAAAAAPxEKEyQtemadhIoehFxU4c0nSCw"></div>
```

Figure 57. viewCreateUser.php

And now the google reCAPTCHA works again shown in Figure 58.



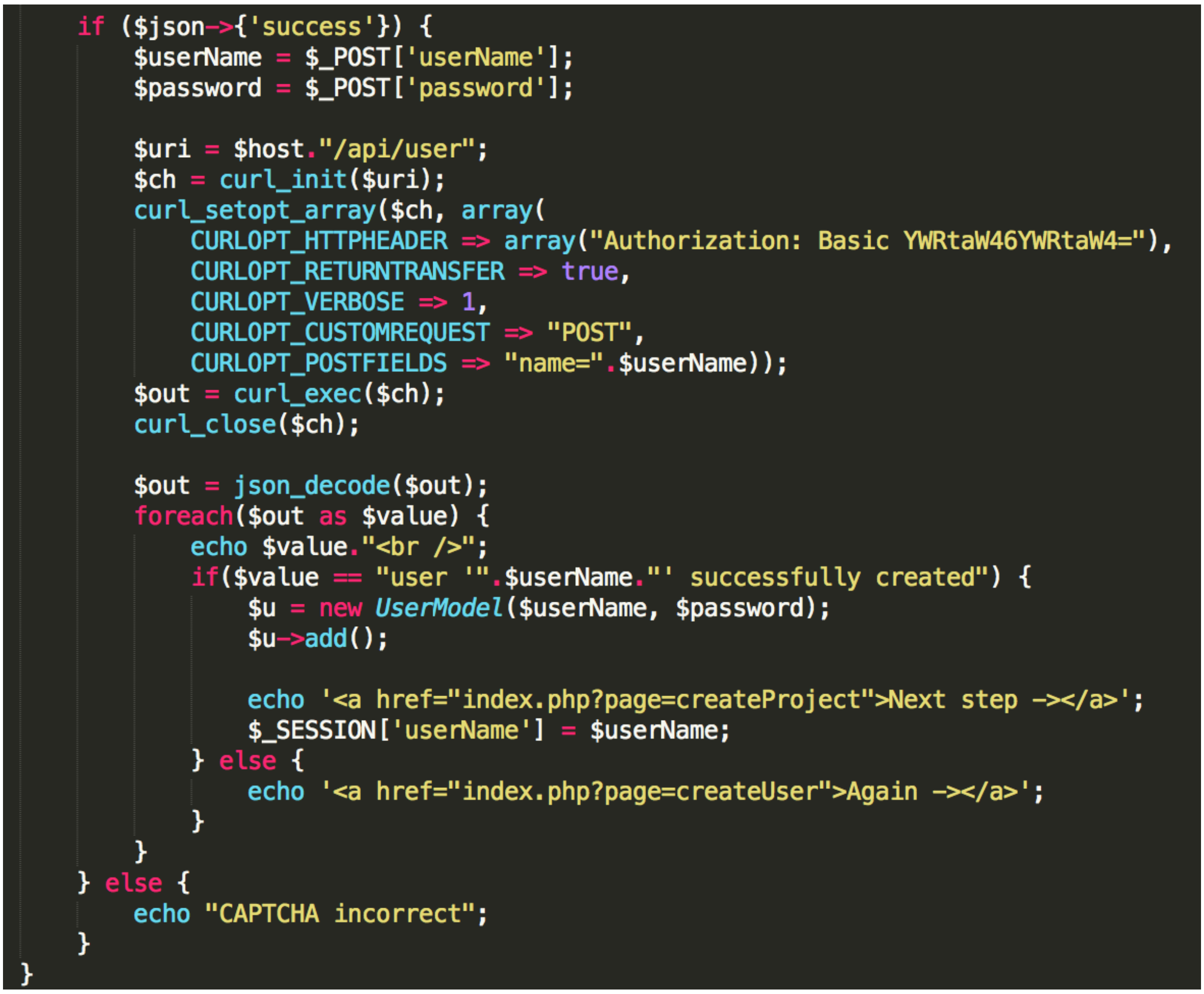
Create User

User name * deng

Password * ••••

Figure 58. success google reCAPTCHA

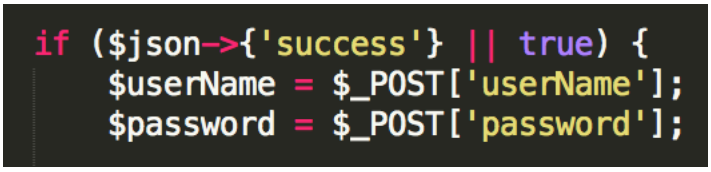
After click “Submit” button, the statement “CAPTCHA incorrect” always return to me, which is another exception. We just go insight to the PHP codes of viewCreatedUser.php. In the following piece of codes:



```php
    if ($json->{'success'}) {
        $userName = $_POST['userName'];
        $password = $_POST['password'];

        $uri = $host."/api/user";
        $ch = curl_init($uri);
        curl_setopt_array($ch, array(
            CURLOPT_HTTPHEADER => array("Authorization: Basic YWRtaW46YWRtaW4="),
            CURLOPT_RETURNTRANSFER => true,
            CURLOPT_VERBOSE => 1,
            CURLOPT_CUSTOMREQUEST => "POST",
            CURLOPT_POSTFIELDS => "name=".$userName));
        $out = curl_exec($ch);
        curl_close($ch);

        $out = json_decode($out);
        foreach($out as $value) {
            echo $value."<br />";
            if($value == "user '".$userName."' successfully created") {
                $u = new UserModel($userName, $password);
                $u->add();

                echo '<a href="index.php?page=createProject">Next step -></a>';
                $_SESSION['userName'] = $userName;
            } else {
                echo '<a href="index.php?page=createUser">Again -></a>';
            }
        }
    } else {
        echo "CAPTCHA incorrect";
    }
}
```

Figure 59. CAPTCHA incorrect

If the statements in the "if condition" equals to "true", then the "CAPTCHA incorrect" will not appear forever. Therefore, we just modify the codes like the following way:



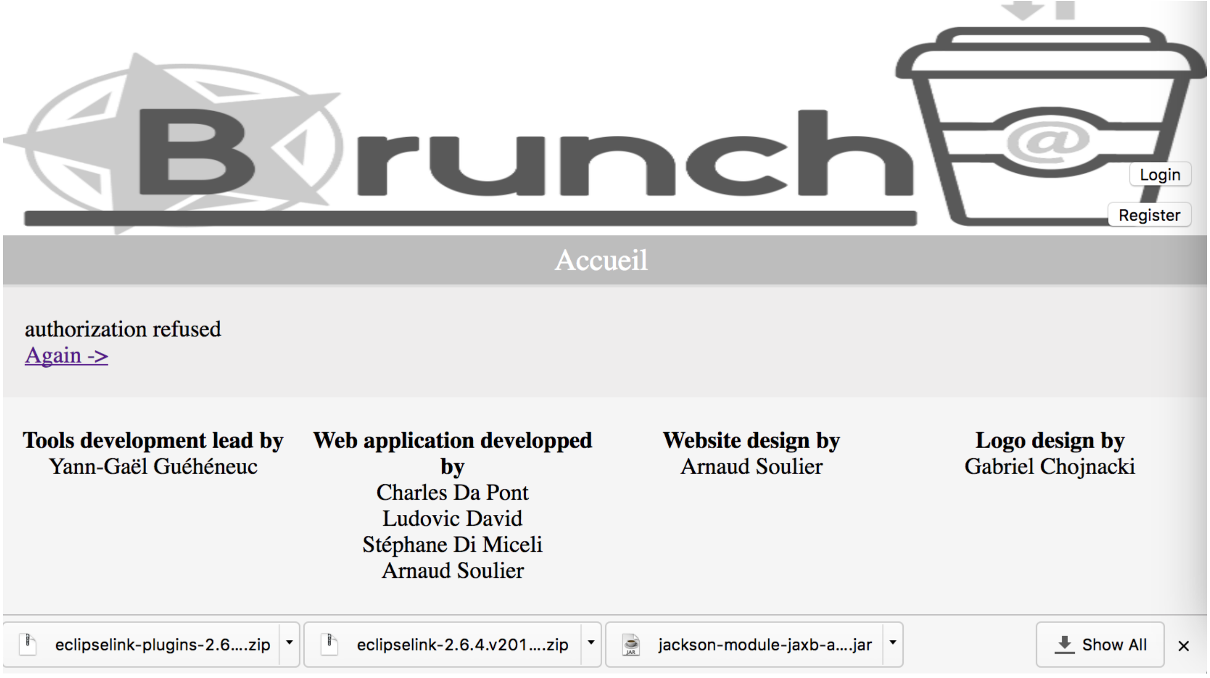
```php
if ($json->{'success'} || true) {
    $userName = $_POST['userName'];
    $password = $_POST['password'];
```

Figure 60. modification of codes

Finally, all the exceptions regarding the google reCAPTCHA has been solved completely.

5.5 Pass the Authorization Refused

After we passed the google reCAPTCHA, there is another exception page in front of us called "authorization refused" as shown in Figure 61.



Accueil

authorization refused
Again ->

Tools development lead by
Yann-Gaël Guéhéneuc

Web application developped by
Charles Da Pont
Ludovic David
Stéphane Di Miceli
Arnaud Soulier

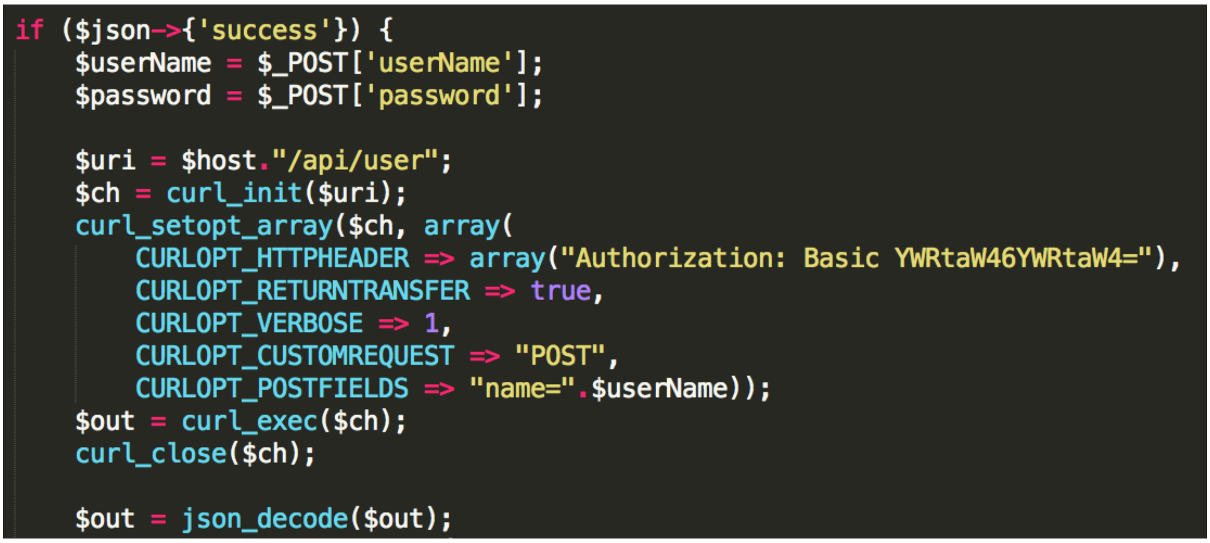
Website design by
Arnaud Soulier

Logo design by
Gabriel Chojnacki

eclipselink-plugins-2.6....zip
eclipselink-2.6.4.v201....zip
jackson-module-jaxb-a....jar
Show All
×

Figure 61. authorization refused

In order to pass the authorization refused, we just go insight the viewCreatedUser.php. The Figure 62 shows the related pieces of codes.



```php
if ($json->{'success'}) {
    $userName = $_POST['userName'];
    $password = $_POST['password'];

    $uri = $host."/api/user";
    $ch = curl_init($uri);
    curl_setopt_array($ch, array(
        CURLOPT_HTTPHEADER => array("Authorization: Basic YWRtaW46YWRtaW4="),
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_VERBOSE => 1,
        CURLOPT_CUSTOMREQUEST => "POST",
        CURLOPT_POSTFIELDS => "name=".$userName));
    $out = curl_exec($ch);
    curl_close($ch);

    $out = json_decode($out);
```

Figure 62. http basic authorization

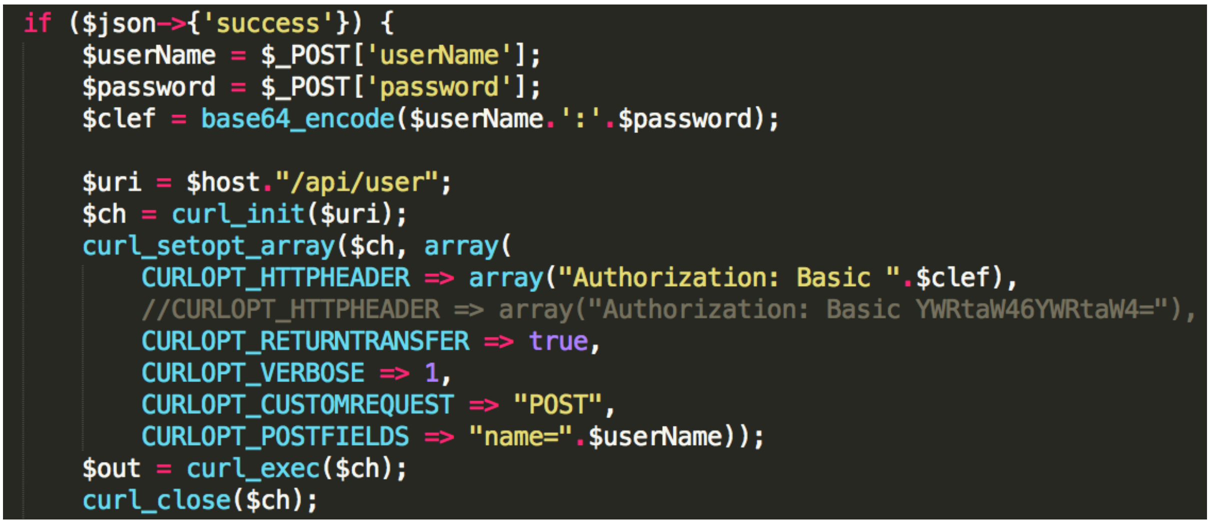
From the beginning, the user is prompted to input their username and the password and these two values are stored as $userName and $password.

Then, initialize a new session and return a cURL handle for use with the curl_setopt(), curl_exec(), and curl_close() functions. After that, curl_setopt() sets multiple options for a cURL session, which statement is useful when set a lot of options instead of calling multiple times. At last, the statement curl_exec() performs the cURL session. The problem occurs in the statement of basic authorization.

After checking the documentations of HTTP header authorization, we can the syntax of the HTTP basic authorization, which is: Authorization: <type> <credentials>. Here, the type is "Basic". And the "credentials" should be constructed like:

- The username and the password are connected with a colon in the middle.
- The result string is base64 encoded.

Therefore, the pieces of codes should be modified like shown in Figure 63.



```
if ($json->{'success'}) {
    $userName = $_POST['userName'];
    $password = $_POST['password'];
    $clef = base64_encode($userName.':'.$password);

    $uri = $host."/api/user";
    $ch = curl_init($uri);
    curl_setopt_array($ch, array(
        CURLOPT_HTTPHEADER => array("Authorization: Basic ".$clef),
        //CURLOPT_HTTPHEADER => array("Authorization: Basic YWRtaW46YWRtaW4="),
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_VERBOSE => 1,
        CURLOPT_CUSTOMREQUEST => "POST",
        CURLOPT_POSTFIELDS => "name=".$userName));
    $out = curl_exec($ch);
    curl_close($ch);
```

Figure 63. modified http basic authorization

And the exception is solved successfully.

5.6 Could not initialize class

The exception is: java.lang.NoClassDefFoundError: Could not initialize class org.eclipse.persistence.jaxb.BeanValidationHelper.

This exception is solved by adding some forgotten JAR in the /module folder of Glassfish. The JAR is https://mvnrepository.com/artifact/com.fasterxml.jackson.module/jackson-module-jaxb-annotations/2.5.1, which contains the same version of the JARs of Jackson. Therefore, I deleted the OSGI-CACHE and put this forgotten JAR to the folder /modules of Glassfish. At last, after restarting the glassfish server, all the exceptions are solved.

5.7 Permission denied

The exception is: Shutting down server due to startup exception
java.net.SocketException: Permission denied.

Some solutions are:

Solution 1

1.1 check if you have another server running or anything that uses the glassfish port
1.2 end all java tasks
1.3 restart glassfish server

Solution 2
Disable firewall
Solution 3
Disable IPV6 by adding some JVM option: -Djava.net.preferIPv4Stack=true
glassfish/domains/domain1/config/domain.xml
<jvm-options>-Djava.net.preferIPv4Stack=true</jvm-options>
Solution 4
run glassfish with a non-root user
set my glassfish to run on port 80 lower than 1024
Only root (sudo) can use port number < 1024

At the last, this exception is solved by issuing the "sudo" command at the beginning of starting the glassfish server.

Chapter 6. Conclusion

Through this internship, my work is basically to improve a web application called "brunch" which is a web application to analyze and evaluate some software systems. I mainly work on some web application programming and database programming. My skills improved in some areas:

- Get some research knowledge including the design patterns.
- Get more familiar with the object-oriented programming, especially the java programming.
- Get some experiences of web application development, especially the PHP programming.
- Get some experiences of database programming, especially the MySQL programming.
- Improve my ability to deal with different exceptions and bugs encountered in the software development.
- Improve my ability to solve the problems by myself during the development.

References

[1] The official Ptidej web site: http://www.ptidej.net/
[2] https://en.wikipedia.org/wiki/Prolog
[3] http://wiki.ptidej.net/doku.php?id=sad
[4] https://www.tutorialspoint.com/design_pattern/design_pattern_overview.htm
[5] file:///Users/dengzichao/Desktop/arnaud/Documents/glassfish4/glassfish/docs/quickstart.html
[6] http://quercus.caucho.com/
[7] https://www.google.com/recaptcha/intro/