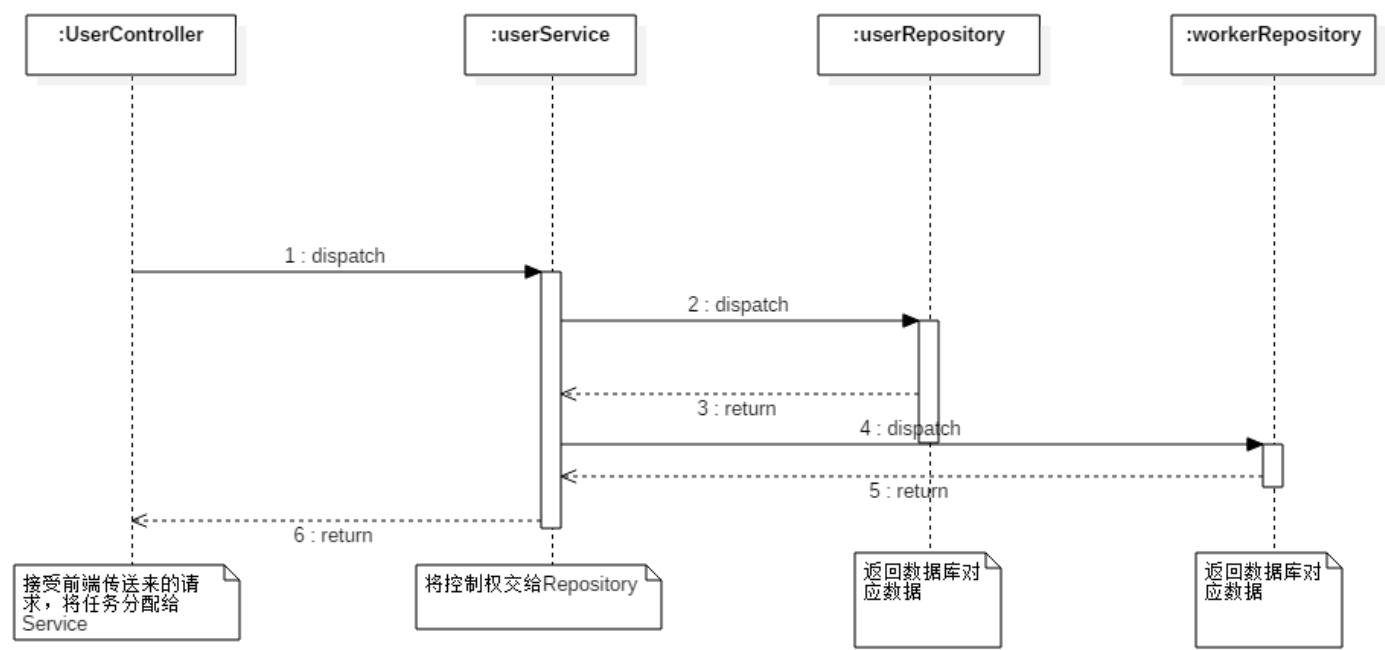


Detailed Design

Process Flow Design

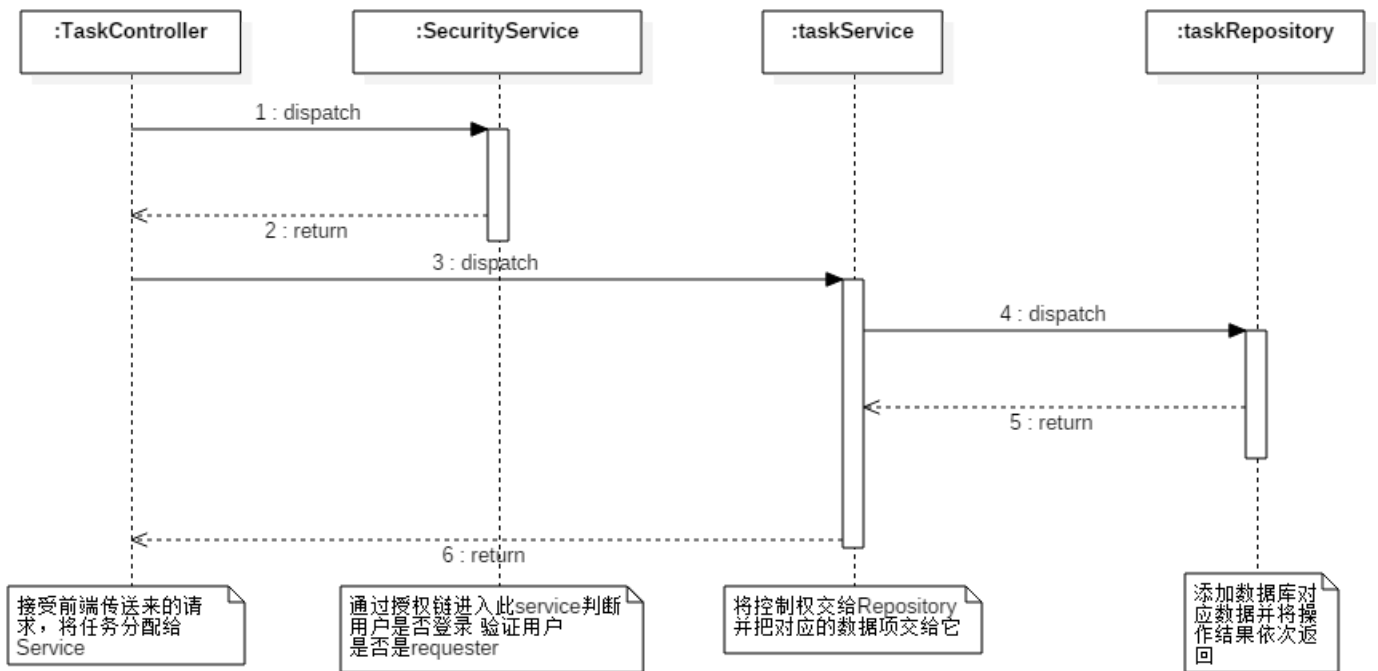
The sequence diagram is required, to show the whole process of the system from the entry point to the end point, for each user scenario. Here are our sequence diagram:

Register(use worker as example)



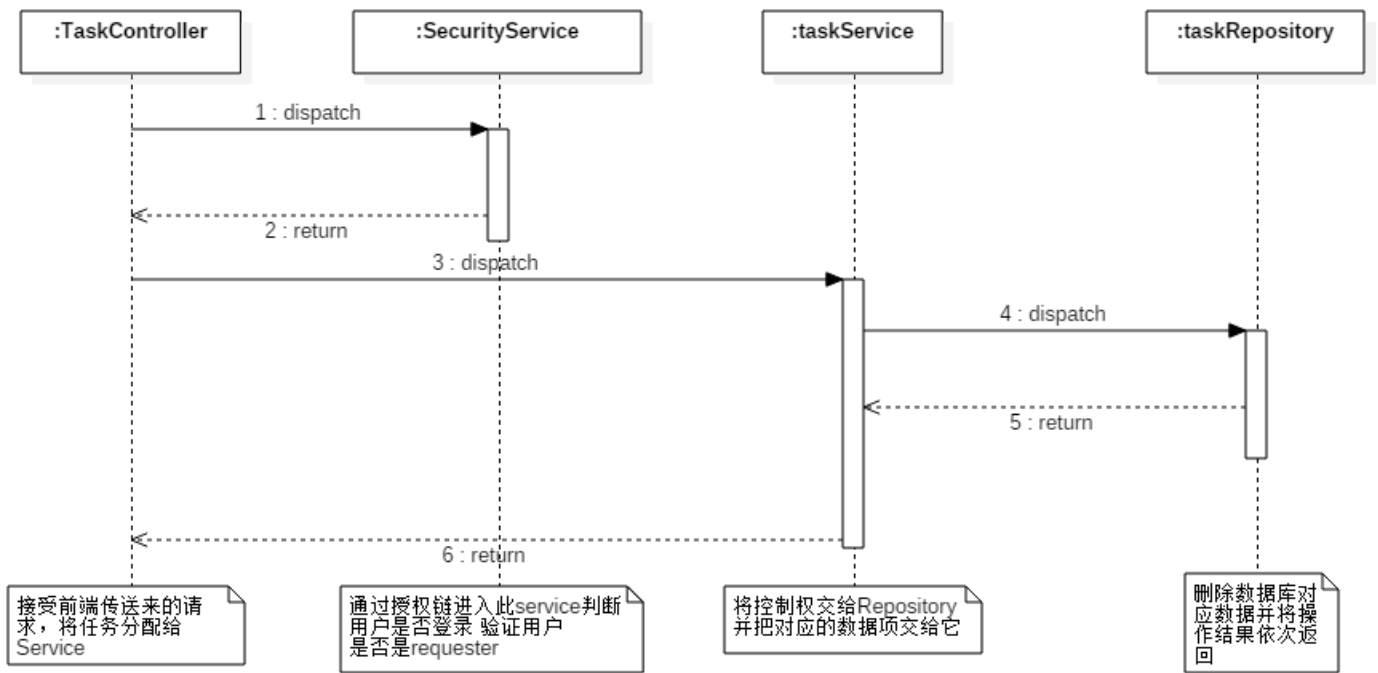
When a user send a request to register, service and repository will help to create a new worker and return the result of the action to the user.

Create Task



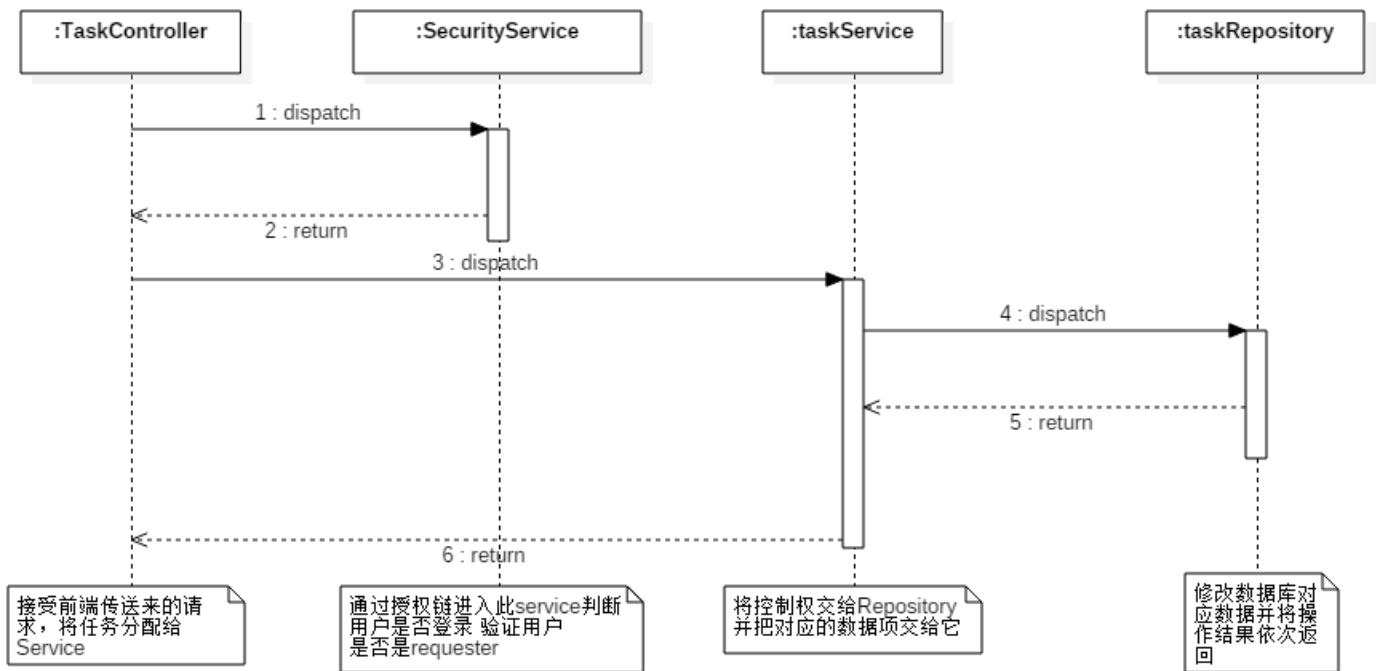
When a user send a request to create a new task, the security service will help to make sure that this user is a requester.If everything is ok, service and repository will do the insertion and return the result of the action to the user.

Delete Task



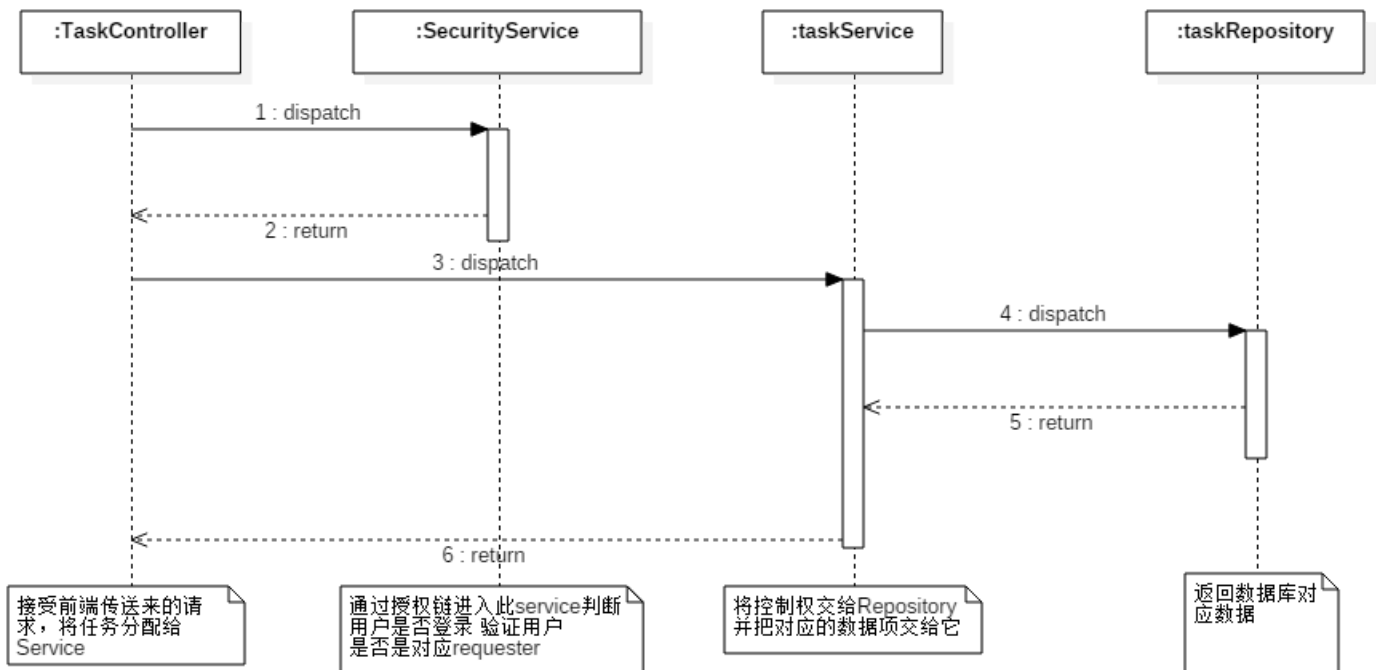
When a user send a request to delete a task, the security service will help to make sure that this user is a corresponding requester.If everything is ok, service and repository will do the deletion and return the result of the action to the user.

Modify Task



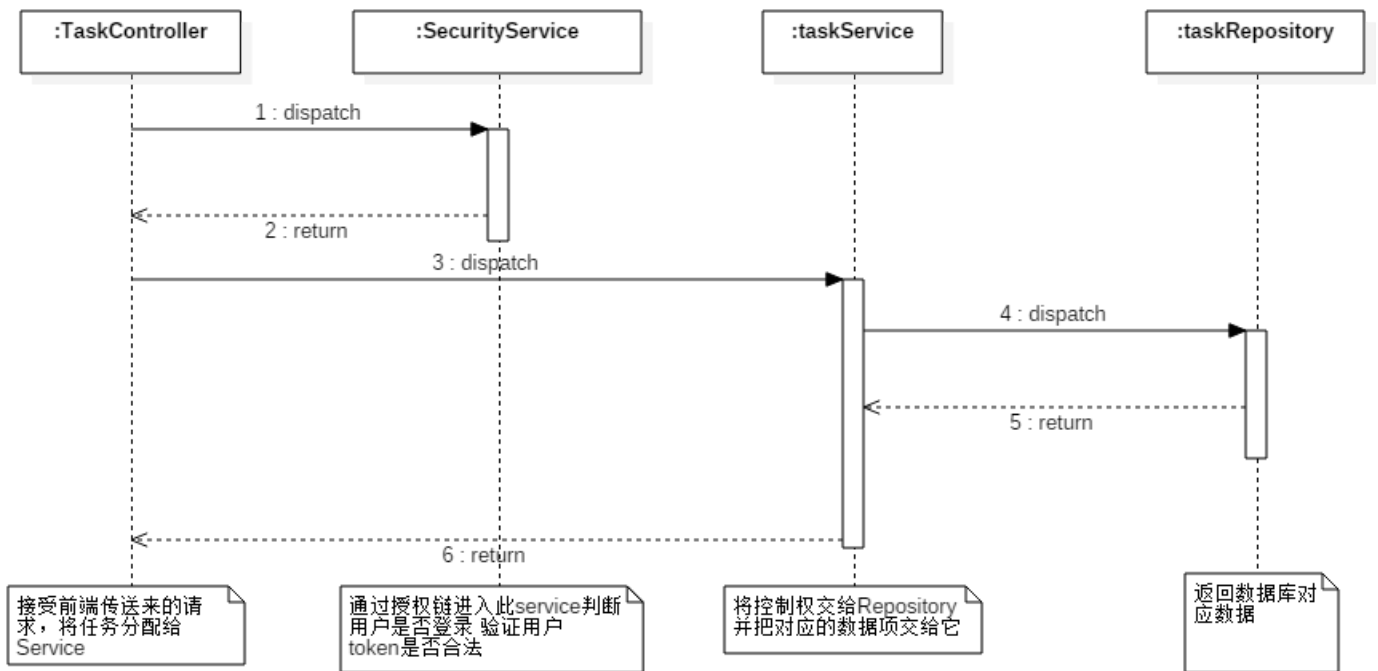
When a user send a request to modify a task, the security service will help to make sure that this user is a corresponding requester.If everything is ok, service and repository will do the modification and return the result of the action to the user.

View Task Data



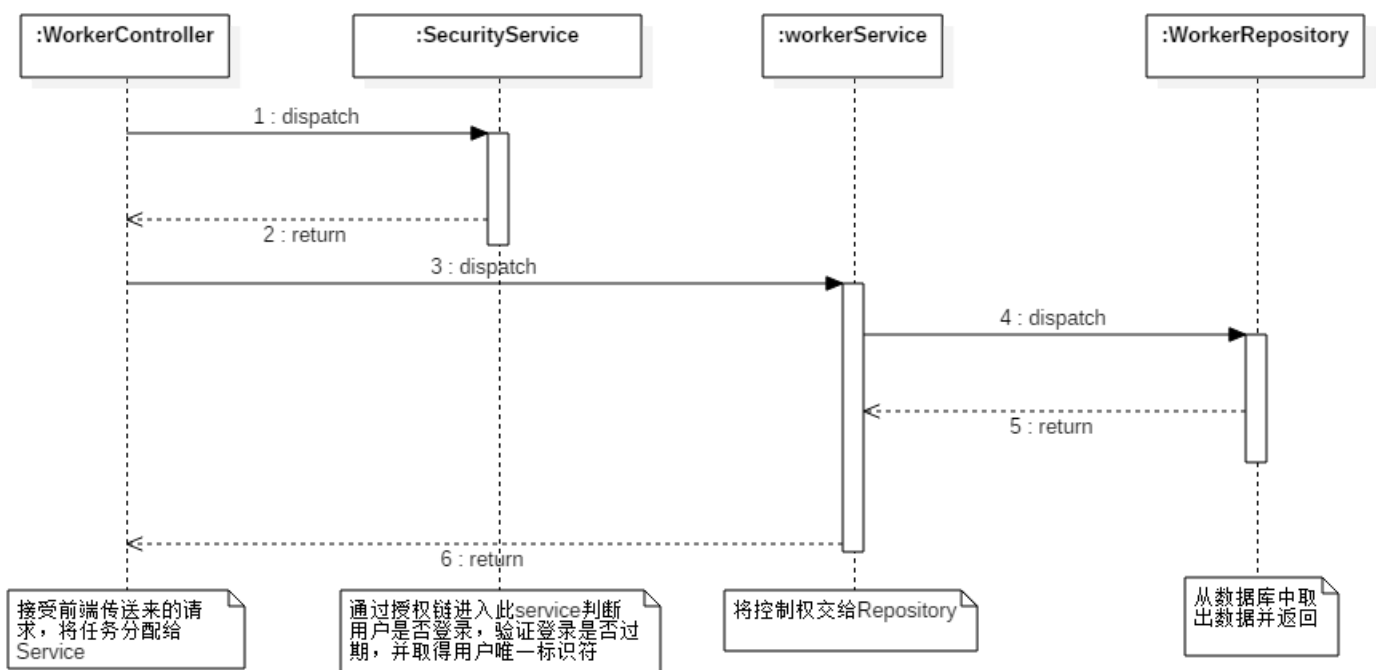
When a user send a request to view the data of a task, the security service will help to make sure that this user is a corresponding requester.If everything is ok, service and repository will return the information to the controller, and finally to the user.

View Task Detailed



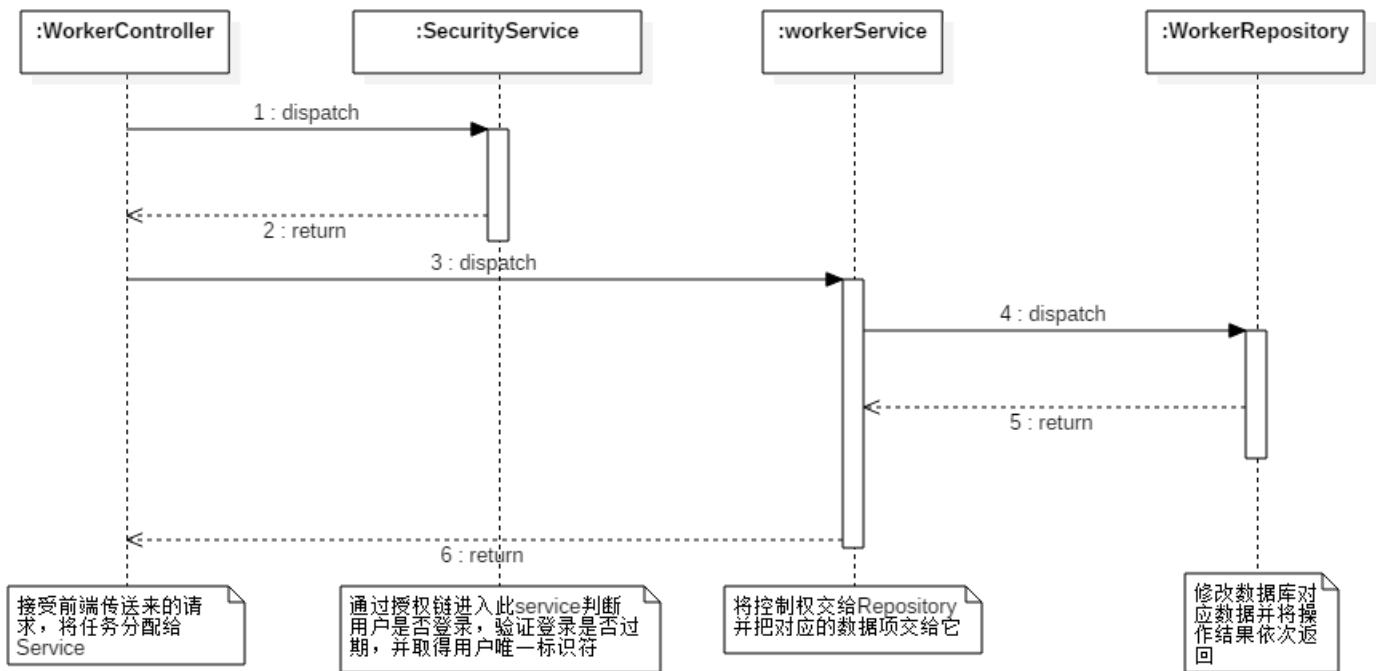
When a user send a requester to view the detailed information of a task, the configuration of security will help us to verify the qualification of the user by examining the token. If everything is ok, service and repository will return the information to the controller, and finally to the user.

View Personal Information(use worker as example)



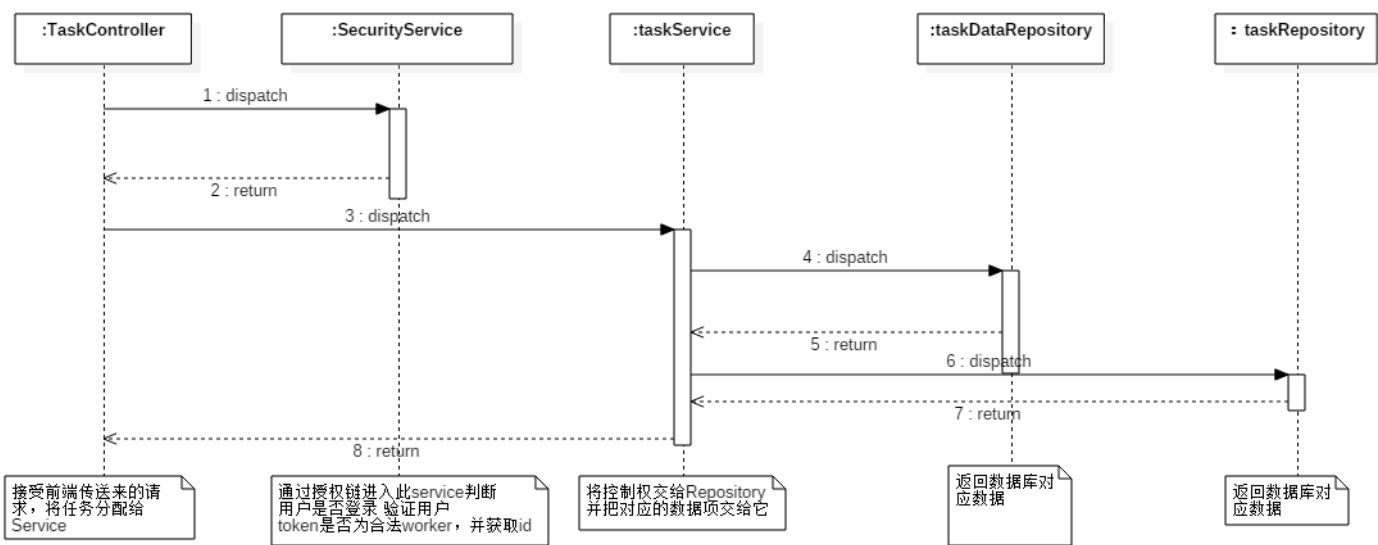
When a user send a requester to view his or her information, the configuration of security will help us to verify the qualification of the user by examining the token. If everything is ok, service and repository will return the information to the controller, and finally to the user.

Update Personal Information(use worker as example)



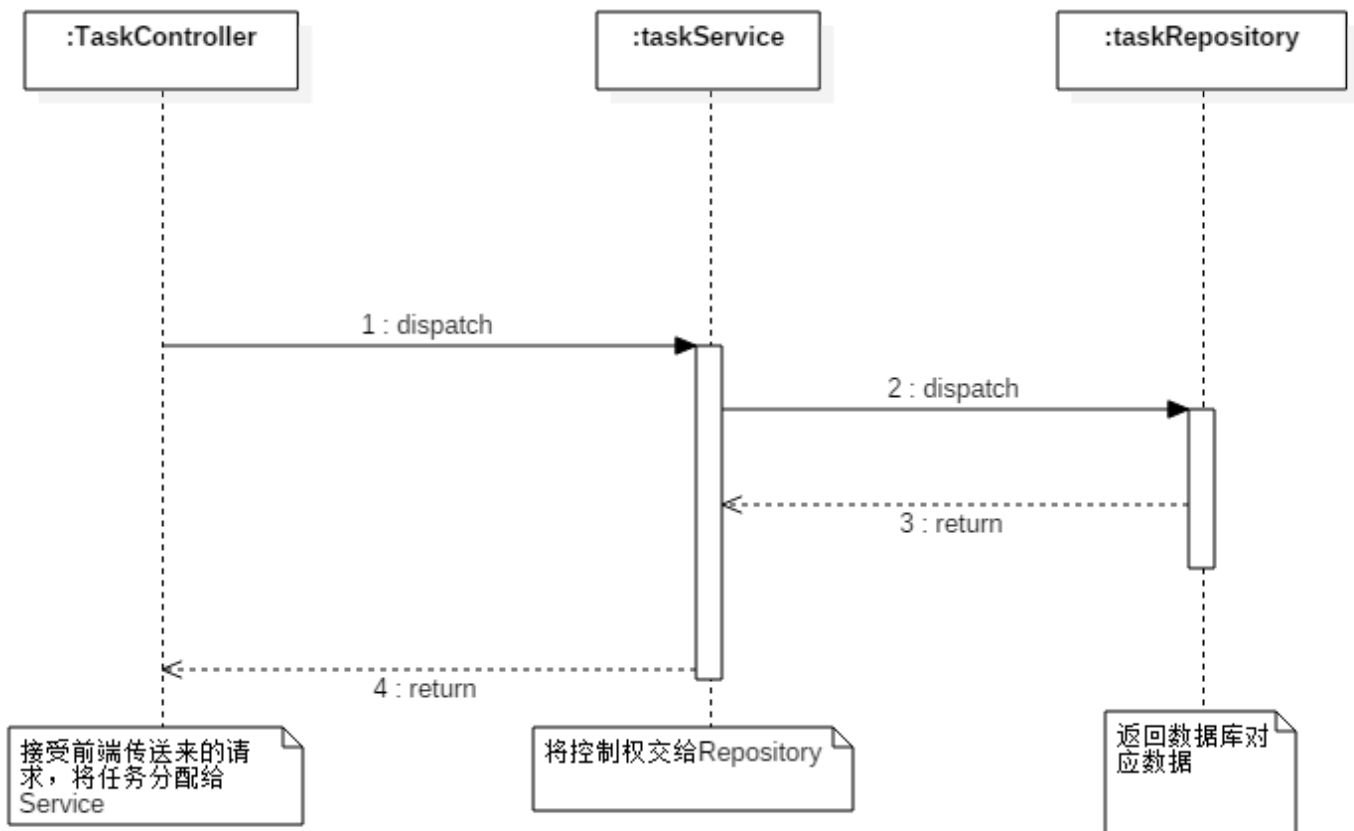
When a user send a requester to updata his or her information, the configuration of security will help us to verify the qualficetion of the user by examing the token. If everything is ok, service and repository will do the updating and return the result of the action to the user.

View Personal Task



When a user send a requester to view the tasks in which he or she get invovled, the configuration of security will help us to verify the qualficetion of the user by examing the token. If everything is ok, service and repository will return the information to the controller, and finally to the user.

Get Task List



When a user send a requester to view the tasks, service and repository will return the information to the controller, and finally to the user.

Algorithm Design

Role Authentication

Background

In this platform, every user has different accout. Some are workers and some are requesters. Tasks are related to money and the information of users and tasks should be kept in secret, so the security is very important. We use spring security combined with JWT(jason web token).

Algorithm Framework(flowchart of pseudocde)



Key Steps

1.Send Request

The program will first check the request sent from the frontend, and check the SecurityConfiguration to make sure whether the request needs a token. If it does, enter the next steps.

2. Do Filter

The program will do the authenticationTokenFilter, it will verify that this request is sent with token in its header and make sure it is legal and not out of date.

3.Verify the Role

Then the spring security will acquire the role of the user, and verify that whether or not this role is able to do this action.

3.Verify the Correspondence

For example, when a user sends a request to update the information of a task, the program will make sure that this user is a requester and he or she is the creator of this task.

Level Calculation

Background

In our platform, requester can set a value to identify whether the worker has permission to accept the task. We call this value level. We do not store the value in the database. We calculate it with exp(experience point), which is stored in database.

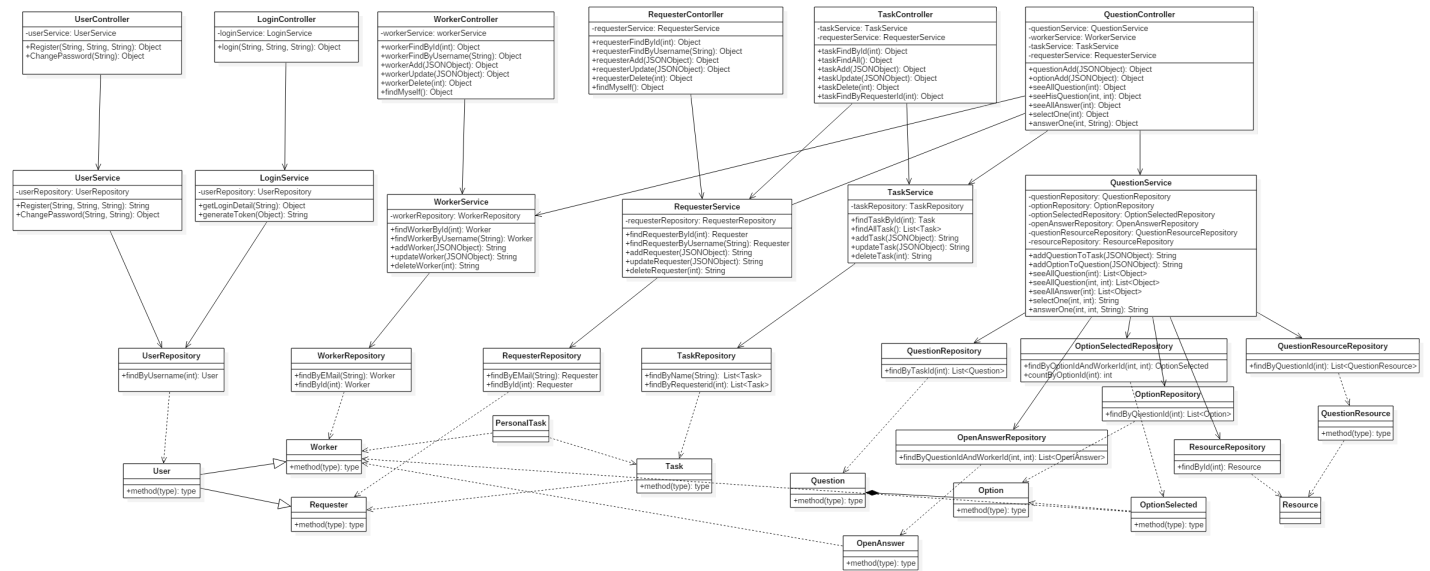
Calculation Way

We consider Overwatch as reference, use the summation function of arithmetic sequence. And get the relation between exp and level with inverse derivation.

$$level = \frac{1 \pm \sqrt{1 + 2 \times exp \div 75}}{2}$$

Class Design

Here is the total class diagram of our project



We use JPA to realize the connection between the backend and the database, so most entity classes' attribute is as same as database design, and the almost all methods of them are setter or getter or constructor, that's why we omit them.

Here gives class description sample:

User

lab.crowdsourcing.entity

*public class **User** implements **UserDetails**, **TokenDetail***

The user instance in the system.

Fields

Modifier and Type	Field	Description
<i>private String</i>	username	the username(email) of the user
<i>private String</i>	password	the password of the user
<i>private String</i>	role	the role of the user

Methods(Constructor and Getter and Setter are omitted)

getAuthorities

- public Collection<? extends GrantedAuthority> getAuthorities()*

Get the authorities of the user.

returns

the authorities of the user

parameters

none

Worker

lab.crowdsourcing.entity

public class **Worker**

The worker instance of the system.

Fields

Modifier and Type	Field	Description
<i>private int</i>	id	the id of the worker
<i>private String</i>	username	the username of the user
<i>private String</i>	name	the true name of the user
<i>private String</i>	teleNumber	the teleNumber of the user
<i>private String</i>	eMail	the eMail of the worker
<i>private int</i>	balance	the balance of the worker
<i>private String</i>	withdrawnMethod	the withdrawnMethod of the worker
<i>private String</i>	education	the education situation of the worker
<i>private String</i>	workArea	the workArea of the worker
<i>private int</i>	age	the age of the worker
<i>private int</i>	level	the level of the worker
<i>private String</i>	gender	the gender of the worker
<i>private String</i>	major	the major of the worker

Methods(Constructor and Getter and Setter are omitted)

setAll

- *public setAll(String username, String name, String teleNumber, String eMail, String withdrawnMethod, String education, String workArea, int age, String gender, String major)*

Set all the information of a worker.

returns

void

parameters

-username:the username of the user
-name:the true name of the user
-teleNumber:the teleNumber of the user
-email:the eMail of the worker
-balance:the balance of the worker
-withdrawnMethod:the withdrawnMethod of the worker
-education:the education situation of the worker
-workArea:the workArea of the worker
-age:the age of the worker
-level:the level of the worker
-gender:the gender of the worker
-major:the major of the worker

Requester

lab.crowdsourcing.entity

*public class **Requester***

The requester instance of the system.

Fields

Modifier and Type	Field	Description
<i>private int</i>	requesterId	the id of the requester
<i>private String</i>	username	the username of the requester
<i>private String</i>	name	the true name of the requester
<i>private String</i>	teleNumber	the teleNumber of the requester
<i>private String</i>	eMail	the eMail of the requester
<i>private int</i>	balance	the balance of the worker
<i>private String</i>	research_field	the research field of the requester
<i>private String</i>	institutionName	the institutionName of the requester

Modifier and Type	Field	Description
<i>private String</i>	address	the address of the requester
<i>private int</i>	age	the age of the requester
<i>private String</i>	gender	the gender of the requester
<i>private String</i>	payMethod	the payMethod of the worker

Methods(Constructor and Getter and Setter are omitted)

setAll

- *public setAll(String username, String name, String teleNumber, String eMail, String research_field, String institutionName, String address, String payMethod, String gender, int age)*

Set all the information of a requester.

returns

void

parameters

-username:the username of the requester
 -name:the true name of the requester
 -teleNumber:the teleNumber of the requester
 -email:the eMail of the requester
 -balance:the balance of the requester
 -research_field:the research field of the requester
 -institutionName:the institutionName of the requester
 -address:the address of the requester
 -age:the age of the requester
 -gender:the gender of the requester
 -payMethod:the payMethod of the requester

Task

lab.crowdsourcing.entity

public class Task

The task instance of the system.

Fields

Modifier and Type	Field	Description
<i>private int</i>	id	the id of the task
<i>private String</i>	name	the name of the task
<i>private String</i>	description	the description of the task
<i>private int</i>	reward	the reward of the task
<i>private String</i>	status	the status of the task
<i>private int</i>	requesterid	the requesterid of the task
<i>private String</i>	type	the type of the task
<i>private String</i>	restrictions	the restrictions of the task
<i>private Timestamp</i>	start_time	the start time of the task
<i>private Timestamp</i>	end_time	the end_time of the task
<i>private int</i>	population	the population of the task
<i>private int</i>	level	the level of the task
<i>private int</i>	time_limitation	the limitation of the time of the task
<i>private int</i>	pay_time	the time of paying of the task

Methods(Constructor and Getter and Setter are omitted)

setAll

- *public setAll(String name, String description, Integer reward, String status, Integer requesterid, String type, String restrictions, Timestamp start_time, Timestamp end_time, int level, int time_limitation, int pay_time)*

Set all the information of a task.

returns

void

parameters

-name:the true name of the task
 -description:the description of the task
 -reward:the reward of the worker
 -status:the status of the task
 -requesterid:the requesterid of the task
 -type:the type of the task
 -restrictions:the restrictions of the task
 -start_time:the start time of the task

-end_time:the end time of the task
-population:the population of the task
-level:the level of the task
-time_limitation:the limitation of the time of the task
-pay_time:the time of paying of the task

Question

lab.crowdsourcing.entity

*public class **Question***

The question instance of the system.

Fields

Modifier and Type	Field	Description
<i>private int</i>	id	the id of the question
<i>private String</i>	content	the content of the question
<i>private int</i>	resource_loading	the situation of the loading resource of the question
<i>private int</i>	type	the type of the question

Option

lab.crowdsourcing.entity

*public class **Option***

The option instance of the system.

Fields

Modifier and Type	Field	Description
<i>private int</i>	id	the id of the option
<i>private String</i>	content	the content of the option
<i>private int</i>	questionId	the question this option belongs to
<i>private int</i>	openAnswerPermission	the permission of the open answer
<i>private int</i>	optionNumber	the sequence number of the option

OpenAnswer

lab.crowdsourcing.entity

public class **OpenAnswer**

The open answer instance of the system.

Fields

Modifier and Type	Field	Description
<i>private int</i>	id	the id of the open answer
<i>private String</i>	content	the content of the option
<i>private int</i>	questionId	the question this option belongs to
<i>private int</i>	workerId	the id of the worker who answered this
<i>private Timestamp</i>	time	time when the worker answer it
<i>private int</i>	optionId	the option this answer belongs to

Resource

lab.crowdsourcing.entity

public class **Resource**

The resource instance of the system.

Fields

Modifier and Type	Field	Description
<i>private int</i>	id	the id of the resource
<i>private String</i>	link	the link of the resource
<i>private String</i>	type	the type of the resource

PhotoAlbum

lab.crowdsourcing.entity

public class **PhotoAlbum**

The photo album instance of the system.

Fields

Modifier and Type	Field	Description
<i>private int</i>	id	the id of the album
<i>private int</i>	requesterId	the id of the creator of the album
<i>private String</i>	name	the name of the album

QuestionResource

lab.crowdsourcing.entity

public class **QuestionResource**

The relationship set between the question and the resource.

Fields

Modifier and Type	Field	Description
<i>private int</i>	id	the id of the set
<i>private int</i>	questionId	the id of the question
<i>private int</i>	resourceId	the id of the resource

PersonalTask

lab.crowdsourcing.entity

public class **PersonalTask**

The relationship set between the task and the worker.

Fields

Modifier and Type	Field	Description
<i>private PersonalTaskKey</i>	id	the id of the set

PersonalTaskKey

lab.crowdsourcing.entity

public class **PersonalTaskKey** implements **Serializable****

The key of the PersonalTask.

Fields

Modifier and Type	Field	Description
private int	workerId	the id of the worker
private int	taskId	the id of the task

Methods(Constructor and Getter and Setter are omitted)

toString

- public String toString()

convert the information to String to implements serializable.

returns

String

parameters

null

OptionSelected

lab.crowdsourcing.entity

public class **OptionSelected**

The option that one worker selected.

Fields

Modifier and Type	Field	Description
private int	id	the id of the instance

Modifier and Type	Field	Description
<i>private int</i>	optionId	the id of the option
<i>private int</i>	workerId	the id of the worker
<i>private Timestamp</i>	time	the time when this worker select the option

ResultMap

lab.crowdsourcing.entity

*public class **ResultMap** extends **HashMap<String, Object>***

The result retrun to the frontend.

Fields

Modifier and Type	Field	Description
<i>private int</i>	id	the id of the instance
<i>private int</i>	optionId	the id of the option
<i>private int</i>	workerId	the id of the worker
<i>private Timestamp</i>	time	the time when this worker select the option

Methods(Constructor and Getter and Setter are omitted)

success

- *public ResultMap success(String code)*

return the success code

returns

ResultMap

parameters

-code: the status code

fail

- *public ResultMap fail(String code)*

return the fail code

returns

ResultMap

parameters

-code: the status code

message

- *public ResultMap message(String code)*

return the result with message

returns

ResultMap

parameters

-message: the message needed to pass

data

- *public ResultMap data(String key, Object obj)*

return the result with data

returns

ResultMap

parameters

-key: the key of the data(key-value)

-obj: the value of the data(key-value)

token

- *public ResultMap token(String token)*

return the result with token

returns

ResultMap

parameters

-token: the token we want to return

Because we use JPA to realize repository, so we omitte them.

LoginService

lab.crowdsourcing.service

_public class **LoginService** _

The service to realize the login.

Fields

Modifier and Type	Field	Description
<i>private UserRepository</i>	userRepository	the repisitory of the user
<i>private TokenUtils</i>	tokenUtils	the util of the token

Methods

getLoginDetail

- *public UserDetails getLoginDetail(String username)*

return the details about user

returns

UserDetails

parameters

-username: the username of the user

generateToken

- *public String generateToken(TokenDetail tokenDetail)*

return the token created by util

returns

String

parameters

-tokenDetail: the details of the user

UserService

lab.crowdsourcing.service

_public class **UserService** _

The service to realize the register and others.

Fields

Modifier and Type	Field	Description
<i>private UserRepository</i>	userRepository	the repisitory of the user

Methods

register

- public String register(String username, String password, String role)*

register for the user

returns

String

parameters

- username: the username of the user
- password: the password of the user
- role:th role of the user

changePassword

- public String changePassword(String name, String password)*

change the password of the user

returns

String

parameters

-username: the username of the user
-password: the new password of the user

WorkerService

lab.crowdsourcing.service

`_public class WorkerService _`

The service to deal with the affair about the worker.

Fields

Modifier and Type	Field	Description
<i>private WorkerRepository</i>	workerRepository	the repository of the worker

Methods

findWorkerById

- *public Worker findWorkerById(int id)*

find a worker by id

returns

Worker

parameters

-id: the id of the worker

findWorkerByUsername

- *public Worker findWorkerByUsername(String username)*

find a worker by username(email)

returns

Worker

parameters

-username: the username of the worker

addWorker

- *public String addWorker(String username, String name, String teleNumber, String eMail, String withdrawnMethod, String education, String workArea, int age, String gender, String major, IdStore id)*

add a new worker with necessary information

returns

String

parameters

- username:the username of the user
- name:the true name of the user
- teleNumber:the teleNumber of the user
- email:the eMail of the worker
- balance:the balance of the worker
- withdrawnMethod:the withdrawnMethod of the worker
- education:the education situation of the worker
- workArea:the workArea of the worker
- age:the age of the worker
- level:the level of the worker
- gender:the gender of the worker
- major:the major of the worker
- id:the id allocated to the worker

updateWorker

- *public String updateWorker(int workerId,String username, String name, String teleNumber, String eMail,String withdrawnMethod, String education, String workArea, int age,String gender, String major)*

update information of a worker

returns

String

parameters

- workerId:the id of the worker
- username:the username of the user
- name:the true name of the user
- teleNumber:the teleNumber of the user
- email:the eMail of the worker
- balance:the balance of the worker
- withdrawnMethod:the withdrawnMethod of the worker
- education:the education situation of the worker
- workArea:the workArea of the worker
- age:the age of the worker
- level:the level of the worker

-gender:the gender of the worker

-major:the major of the worker

deleteWorker

- *public String deleteWorker(int id)*

delete a worker

returns

String

parameters

-id:the id of the worker

RequesterService

lab.crowdsourcing.service

`_public class RequesterService _`

The service to deal with the affair about the requester.

Fields

Modifier and Type	Field	Description
<i>private RequesterRepository</i>	requesterRepository	the repository of the requester

Methods

findRequesterById

- *public Requester findRequesterById(int id)*

find a worker by id

returns

Requester

parameters

-id: the id of the requester

findRequesterByUsername

- *public Requester findRequesterByUsername(String username)*

find a requester by username(email)

returns

String

parameters

-username: the username(email) of the worker

addRequester

- *addRequester(String username, String name, String teleNumber, String eMail, String research_field, String institutionName, String address, String payMethod, String gender, int age, IdStore id)*

add a requester with necessary information

returns

String

parameters

-username:the username of the requester

-name:the true name of the requester

-teleNumber:the teleNumber of the requester

-email:the eMail of the requester

-balance:the balance of the requester

-research_field:the research field of the requester

-institutionName:the institutionName of the requester

-address:the address of the requester

-age:the age of the requester

-gender:the gender of the requester

-payMethod:the payMethod of the requester

-id:the id allocated to the requester

updateRequester

- *updateRequester(int requesterId,String username, String name, String teleNumber, String eMail, String research_field, String institutionName, String address, String payMethod, String gender, int age)*

update the information of the requester

returns

String

parameters

-requesterId:the id of the requester
-username:the username of the requester
-name:the true name of the requester
-teleNumber:the teleNumber of the requester
-email:the eMail of the requester
-balance:the balance of the requester
-research_field:the research field of the requester
-institutionName:the institutionName of the requester
-address:the address of the requester
-age:the age of the requester
-gender:the gender of the requester
-payMethod:the payMethod of the requester

deleteRequester

- *public String deleteRequester(int id)*

delete a requester

returns

String

parameters

-id:the id of the requester

TaskService

lab.crowdsourcing.service

_public class TaskService _

The service related with task.

Fields

Modifier and Type	Field	Description
<i>private TaskRepository</i>	taskRepository	the repisitory of the task

Methods

findAllTask

- *public List<Task> findAllTask()*

return the details about all the tasks

returns

List<Task>

parameters

null

generateToken

- *public Task findTaskById(int id)*

find a task by its id

returns

Task

parameters

-id: the id of the task

findTaskByRequesterId

- *public List<Task> findTaskByRequesterId(int id)*

find tasks by the id of its publisher

returns

List<Task>

parameters

-id: the id of the requester

addTask

- *public String addTask(String name, String description, Float reward, String status, Integer requesterid, String type, String restrictions, Timestamp start_time, Timestamp end_time,int level, Float time_limitation, Float pay_time,String area,String usage,int min_age,int max_age,IdStore taskId)*

add a task with the necessary information

returns

String

parameters

-name:the true name of the task
-description:the description of the task
-reward:the reward of the worker
-status:the status of the task
-requesterid:the requesterid of the task
-type:the type of the task
-restrictions:the restrictions of the task
-start_time:the start time of the task
-end_time:the end time of the task
-population:the population of the task
-level:the level of the task
-time_limitation:the limitation of the time of the task
-pay_time:the time of paying of the task
-taskId:the id allocated to the task

updateTask

- *public String updateTask(int taskId,String name, String description, Float reward, String status, Integer requesterid, String type, String restrictions, Timestamp start_time, Timestamp end_time,int level, Float time_limitation, Float pay_time,String area,String usage,int min_age,int max_age)*

update the information of the task

returns

String

parameters

-name:the true name of the task
-description:the description of the task
-reward:the reward of the worker
-status:the status of the task
-requesterid:the requesterid of the task
-type:the type of the task
-restrictions:the restrictions of the task
-start_time:the start time of the task
-end_time:the end time of the task
-population:the population of the task
-level:the level of the task
-time_limitation:the limitation of the time of the task
-pay_time:the time of paying of the task
-taskId:the id allocated to the task

deleteTask

- *public String deleteTask(int id)*

delete a task

returns

String

parameters

-id:the id of the task

PersonalTaskService

lab.crowdsourcing.service

`_public class PersonalTaskService _`

The service about the relationship between the worker and task

Fields

Modifier and Type	Field	Description
<i>private TaskRepository</i>	taskRepository	the repository of the task
<i>private WorkerRepository</i>	workerRepository	the repository of the worker
<i>private PersonalTaskRepository</i>	personalTaskRepository	the repository of the personalTask

Methods

findTaskByWorkerId

- `public List<Task> findTaskByWorkerId(int id)`

find the tasks the specified worker get involved in

returns

List<Task>

parameters

-id:the id of the worker

findWorkerByTaskId

- `public List<Worker> findWorkerByTaskId(int id)`

find the workers who get involve in the specified task

returns

List<Worker>

parameters

-id: the id of the worker

addPersonalTask

- *public String addPersonalTask(Integer workerid, Integer taskid)*

help a worker to get involved in a task

returns

String

parameters

-workerid: the id of the worker

-taskid:the id of the task

deletePersonalTask

- *public String deletePersonalTask(Integer workerid, Integer taskid)*

help a worker to abandon a task

returns

String

parameters

-workerid: the id of the worker

-taskid:the id of the task

PhotoService

lab.crowdsourcing.service

`_public class PhotoService _`

The service to realize the photo library.

Fields

Modifier and Type	Field	Description
<i>private PhotoAlbumRepository</i>	PhotoAlbumRepository	the repository of the photo
<i>private ResourceRepository</i>	resourceRepository	the repository of the resource
<i>private String</i>	location	the location to store pictures

Methods

findPhotos

- *public List<PhotoAlbumDetail> findPhotos(int requesterId)*

find photos and photoAlbum details by requesterId

returns

List<PhotoAlbumDetail>

parameters

-requesterId: the id of the requester

addPhotoAlbum

- *public int addPhotoAlbum(int requesterId,String name)*

The service to create a new photoAlbum.

returns

int

parameters

-requesterId: the id of the requester

-name:the name of the album

changePhotoAlbumName

- *public int changePhotoAlbumName(int id,String name)*

The service to change the name of the album.

returns

int

parameters

-id: the id of the album

-name:the new name of the album

addPhoto

- *public String addPhoto(MultipartFile multipartFile, int photoAlbumId)*

The service to add a picture into a album.

returns

String

parameters

-multipartFile: the picture need to add

-photoAlbumId:the id of the album

deletePhoto

- *public int deletePhoto(int id)*

The service to delete a picture.

returns

int

parameters

-id:the id of the picture

QuestionService

lab.crowdsourcing.service

`_public class QuestionService _`

The service is related to the question.

Fields

Modifier and Type	Field	Description
<i>private QuestionRepository</i>	questionRepository	the repository of the question
<i>private OptionRepository</i>	optionRepository	the repository of the option
<i>private OptionSelectedRepository</i>	optionSelectedRepository	the repository of the optionSelected

Modifier and Type	Field	Description
<i>private OpenAnswerRepository</i>	openAnswerRepository	the repository of the open Answer
<i>private QuestionResourceRepository</i>	QuestionResourceRepository	the repository of questionResource
<i>private ResourceRepository</i>	resourceRepository	the repository of resource

Methods

addQuestionToTask

- *public String addQuestionToTask(int taskId,String content,int resourceLoading,int type,int compulsory,IdStore made)*

add a question to a task

returns

String

parameters

- taskId: the id of the task
- content:the content of the question
- resouceloading:whether the question need resource to load
- type:the type of the question
- compulsory:whether the question is a compulsory question
- made:the id allocated to the question

addOptionToQuestion

- *public String addOptionToQuestion(String content,Integer questionId, Integer openAnswerPermission , Integer optionNumber,IdStore made)*

add a option to a question

returns

String

parameters

- content: the content of the option
- questionId:the id of the question
- openAnswerPermission:the permission of the openAnswer
- optionNumber:the sequence number of the option
- made:the id allocated to the option

seeAllQuestion

- *public List<QuestionDetail> seeAllQuestion(int taskId)*

see all the questions of a task

returns

List<QuestionDetail>

parameters

-taskId:the id of the task

seeAllQuestion

- *public List<QuestionDetail> seeAllQuestion(int taskId,int workerId)*

see all the questions and answers of a task and a worker

returns

List<QuestionDetail>

parameters

-taskId:the id of the task

-workerId:the id of the worker

seeAllAnswer

- *public List<QuestionDetail> seeAllAnswer(int taskId)*

see all the answers of the question

returns

List<QuestionDetail>

parameters

-taskId:the id of the task

selectOne

- *public String selectOne(int optionId,int workerId)*

The service to select a option

returns

String

parameters

- optionId:the id of the option
- workerId:the id of the worker

answerOne

- *public String answerOne(int optionId,int workerId,String content)*

The service to answer a openAnswer question

returns

String

parameters

- optionId:the id of the option
- workerId:the id of the worker
- content:the content of the answer

addResource

- *public int addResource(int questionId,int resourceId)*

The service to add a resouce to a question

returns

int

parameters

- questionId:the id of the question
- resourceId:the id of the resource

LoginController

lab.crowdsourcing.controller

`_public class LoginController _`

The controller to realize the login.

Fields

Modifier and Type	Field	Description
<i>private String</i>	tokenHeader	the header of the token

Modifier and Type	Field	Description
<i>private UserRepository</i>	userRepository	the repository of the user

Methods

checkAccount

- *private ResultMap checkAccount(String username, String password, UserDetails loginDetail,String role)*

help to check the account

returns

ResultMap

parameters

-username: the username of the user
 -password:the password of the user
 -loginDetail:the detail of the user
 -role:the role of the user

login

- *public ResultMap login(String username, String password,String role)*

the interface help to login

returns

ResultMap

parameters

-username: the username of the user
 -password:the password of the user
 -role:the role of the user

UserController

lab.crowdsourcing.controller

`_public class UserController _`

The controller about register.

Fields

Modifier and Type	Field	Description
<i>private String</i>	tokenHeader	the header of the token
<i>private TokenUtils</i>	tokenUtils	the util of the token
<i>private HttpServletRequest</i>	request	the request frontend send
*private UserService *	userService	the service of the user
<i>private RequesterService</i>	requesterService	the service of the requester
<i>private WorkerService</i>	workerService	the service of the worker

Methods

registerAsRequester

- *public ResultMap registerAsRequester(String eMail,String password, String username, String name, String teleNumber, String research_field, String institutionName, String address, String payMethod, String gender, int age)*

help to register as a requester

returns

ResultMap

parameters

-username:the username of the requester
 -name:the true name of the requester
 -teleNumber:the teleNumber of the requester
 -email:the eMail of the requester
 -balance:the balance of the requester
 -research_field:the research field of the requester
 -institutionName:the institutionName of the requester
 -address:the address of the requester
 -age:the age of the requester
 -gender:the gender of the requester
 -payMethod:the payMethod of the requester
 -password:the password of the requester

registerAsWorker

- *public ResultMap registerAsWorker(String eMail,String password, String username, String name, String teleNumber, String withdrawnMethod, String education, String workArea, int age, String gender, String major)*

help to register as a worker

returns

ResultMap

parameters

- username:the username of the user
- name:the true name of the user
- teleNumber:the teleNumber of the user
- email:the eMail of the worker
- balance:the balance of the worker
- withdrawnMethod:the withdrawnMethod of the worker
- education:the education situation of the worker
- workArea:the workArea of the worker
- age:the age of the worker
- level:the level of the worker
- gender:the gender of the worker
- major:the major of the worker
- password:the password of the worker

changePassword

- *public ResultMap changePassword(String password)*

help to change password

returns

ResultMap

parameters

- password:the new password of the user

WorkerController

lab.crowdsourcing.controller

`_public class WorkerController _`

help to find or change information of a worker

Fields

Modifier and Type	Field	Description
<i>private String</i>	tokenHeader	the header of the token
<i>private TokenUtils</i>	tokenUtils	the util of the token

Modifier and Type	Field	Description
<i>private HttpServletRequest</i>	request	the request frontend send
<i>private WorkerService</i>	workerService	the service of the worker

Methods

workerFindByld

- *public ResultMap workerFindByld(int id)*

find a worker by id

returns

ResultMap

parameters

-id:the id of the worker

workerUpdate

- *public ResultMap workerUpdate(String username, String name, String teleNumber, String eMail, String withdrawnMethod, String education, String workArea, int age, String gender, String major)*

help to update a worker's information

returns

ResultMap

parameters

-username:the username of the user
 -name:the true name of the user
 -teleNumber:the teleNumber of the user
 -email:the eMail of the worker
 -balance:the balance of the worker
 -withdrawnMethod:the withdrawnMethod of the worker
 -education:the education situation of the worker
 -workArea:the workArea of the worker
 -age:the age of the worker
 -gender:the gender of the worker
 -major:the major of the worker

findMyself

- *public ResultMap findMyself()*

find oneself information with token

returns

ResultMap

parameters

null

RequesterController

lab.crowdsourcing.controller

`_public class RequesterController _`

help to find or change information of a worker

Fields

Modifier and Type	Field	Description
<i>private String</i>	tokenHeader	the header of the token
<i>private TokenUtils</i>	tokenUtils	the util of the token
<i>private HttpServletRequest</i>	request	the request frontend send
<i>private RequesterService</i>	requesterService	the service of the requester

Methods

requesterFindById

- *public ResultMap requesterFindById(int id)*

find a requester by id

returns

ResultMap

parameters

-id:the id of the requester

requesterUpdate

- *public ResultMap requesterUpdate(String username, String name, String teleNumber, String eMail, String research_field, String institutionName, String address, String payMethod, String gender, int age)*

help to update a requester' information

returns

ResultMap

parameters

- username:the username of the requester
- name:the true name of the requester
- teleNumber:the teleNumber of the requester
- email:the eMail of the requester
- balance:the balance of the requester
- research_field:the research field of the requester
- institutionName:the institutionName of the requester
- address:the address of the requester
- age:the age of the requester
- gender:the gender of the requester
- payMethod:the payMethod of the requester

findMyself

- *public ResultMap findMyself()*

find oneself informtion with token

returns

ResultMap

parameters

null

ImageController

lab.crowdsourcing.controller

_public class ImageController _

help to find or change information of a worker

Fields

Modifier and Type	Field	Description
<i>private String</i>	tokenHeader	the header of the token

Modifier and Type	Field	Description
<i>private TokenUtils</i>	tokenUtils	the util of the token
<i>private HttpServletRequest</i>	request	the request frontend send
<i>private RequesterService</i>	requesterService	the service of the requester
<i>private String</i>	location	the location to store images
<i>private PhotoService</i>	photoService	the service of the image

Methods

addAlbum

- *public ResultMap addAlbum(String name)*

create a new photo album

returns

ResultMap

parameters

-name:the name the album

changeAlbumName

- *public ResultMap changeAlbumName(int id,String name)*

help to change a album name

returns

ResultMap

parameters

-id:the id of the album

-name:the new name of the album

findImages

- *public ResultMap findImages()*

find all the images and albums with token

returns

ResultMap

parameters

null

addImage

- *public ResultMap addImage(MultipartFile multipartFile,int photoAlbumId) throws IOException*

add a new image into a album

returns

ResultMap

parameters

-multipartFile:the image

-photoAlbumId:the id of the album

deletelImage

- *public ResultMap deletelImage(int id)*

delete a image in a album

returns

ResultMap

parameters

-id:the id of the image

TaskController

lab.crowdsourcing.controller

`_public class TaskController _`

help to solve the affair related to task

Fields

Modifier and Type	Field	Description
<i>private String</i>	tokenHeader	the header of the token

Modifier and Type	Field	Description
<i>private TokenUtils</i>	tokenUtils	the util of the token
<i>private HttpServletRequest</i>	request	the request frontend send
<i>private RequesterService</i>	requesterService	the service of the requester
<i>private String</i>	location	the location to store images
<i>private TaskService</i>	taskService	the service of the image

Methods

taskFindAll

- *public ResultMap taskFindAll()*

find all the tasks

returns

ResultMap

parameters

null

taskFindById

- *public ResultMap taskFindById(int id)*

find a task by its id

returns

ResultMap

parameters

-id:the id of the task

taskFindByRequesterId

- *public ResultMap taskFindByRequesterId(int requesterid)*

find tasks by its publisher id

returns

ResultMap

parameters

-requesterid:the id of the requester

taskAdd

- *public ResultMap taskAdd(String name, String description, Float reward, String status, String type, String restrictions, Timestamp start_time, Timestamp end_time, int level, Float time_limitation, Float pay_time,String area,String usage,int min_age,int max_age)*

create a new task

returns

ResultMap

parameters

-name:the true name of the task
-description:the description of the task
-reward:the reward of the worker
-status:the status of the task
-requesterid:the requesterid of the task
-type:the type of the task
-restrictions:the restrictions of the task
-start_time:the start time of the task
-end_time:the end time of the task
-population:the population of the task
-level:the level of the task
-time_limitation:the limitation of the time of the task
-pay_time:the time of paying of the task

taskUpdate

- *public ResultMap taskUpdate(int taskId,String name, String description, Float reward, String status, String type, String restrictions, Timestamp start_time, Timestamp end_time,int level, Float time_limitation, Float pay_time,String area,String usage,int min_age,int max_age)*

update a task's information

returns

ResultMap

parameters

-taskId:the id of the task
-name:the true name of the task
-description:the description of the task

-reward:the reward of the worker
-status:the status of the task
-requesterid:the requesterid of the task
-type:the type of the task
-restrictions:the restrictions of the task
-start_time:the start time of the task
-end_time:the end time of the task
-population:the population of the task
-level:the level of the task
-time_limitation:the limitation of the time of the task
-pay_time:the time of paying of the task

taskDelete

- *public ResultMap taskDelete(int id)*

delete a task

returns

ResultMap

parameters

-id:the id of the task

findMyPublishTask

- *public ResultMap findMyPublishTask()*

find one'own tasks

returns

ResultMap

parameters

null

PersonalTaskController

lab.crowdsourcing.controller

`_public class PersonalTaskController _`

help to solve the affair related to relation between the worker and task

Fields

Modifier and Type	Field	Description
<i>private String</i>	tokenHeader	the header of the token
<i>private TokenUtils</i>	tokenUtils	the util of the token
<i>private HttpServletRequest</i>	request	the request frontend send
<i>private personalTaskService</i>	personalTaskService	the service of the personalTask
<i>private WorkerService</i>	workerService	the service of the worker

Methods

personalTaskFindTaskList

- *public ResultMap personalTaskFindTaskList(int id)*

find all the tasks one worker gets involved in

returns

ResultMap

parameters

-id:the id of worker

personalTaskFindWorkerList

- *public ResultMap personalTaskFindWorkerList(int id)*

find all the workers that get involved in the specific task

returns

ResultMap

parameters

-id:the id of the task

personalTaskAdd

- *public ResultMap personalTaskAdd(Integer taskId)*

help to get involved in one task

returns

ResultMap

parameters

-taskId:the id of the task

personalTaskDelete

- public ResultMap personalTaskDelete(Integer workerId, Integer taskId)*

help a worker to abandon a task

returns

ResultMap

parameters

-workerId:the id of the worker

-taskId:the id of the task

findMyTask

- public ResultMap findMyTask()*

find all the tasks one worker get involved in with token

returns

ResultMap

parameters

null

QuestionController

lab.crowdsourcing.controller

`_public class QuestionController _`

help to solve the affair related to task

Fields

Modifier and Type	Field	Description
<i>private String</i>	tokenHeader	the header of the token

Modifier and Type	Field	Description
<i>private TokenUtils</i>	tokenUtils	the util of the token
<i>private HttpServletRequest</i>	request	the request frontend send
<i>private personalTaskService</i>	personalTaskService	the service of the personalTask
<i>private QuestionService</i>	questionService	the service of the question
<i>private RequesterService</i>	requesterService	the service of the requester
<i>private TaskService</i>	taskService	the service of the task
<i>private WorkerService</i>	workerService	the service of the worker

Methods

questionAdd

- *public ResultMap questionAdd(int taskId,String content,int resourceLoading,int type,int compulsory)*

add a new question

returns

ResultMap

parameters

- taskId:the id of task
- content:the content of the question
- resourceLoading:the situation about the resource
- type:the type of the question
- compulsory:whether the question is a compulsory question

optionAdd

- *public ResultMap optionAdd(String content,Integer questionId, Integer openAnswerPermission , Integer optionNumber)*

add a new option

returns

ResultMap

parameters

- content:the content of the option
- questionId:the id of the question

-openAnswerPermission:the permission of the open answer

-optionNumber:the sequence number of the option

seeAllQuestion

- *public ResultMap seeAllQuestion(int taskId)*

see all the questions of a task

returns

ResultMap

parameters

-taskId:the id of the task

seeMyQuestion

- *public ResultMap seeMyQuestion(int taskId)*

see allthe questions and one's own answers of a tasks

returns

ResultMap

parameters

-taskId:the id of the task

seeHisQuestion

- *public ResultMap seeMyQuestion(int taskId)*

see allthe questions and one's answers of a tasks

returns

ResultMap

parameters

-taskId:the id of the task

-workerId:the id of the worker

seeAllAnswer

- *public ResultMap seeAllAnswer(int taskId)*

find all the answers of a task

returns

ResultMap

parameters

-taskId:the id of the taskId

selectOne

- *public ResultMap selectOne(int optionId)*

select one option

returns

ResultMap

parameters

-optionId:the id of the option

answerOne

- *public ResultMap answerOne(int optionId,String content)*

answer one openAnswer quesrtion

returns

ResultMap

parameters

-optionId:the id of the option

-content:the content of the answer

answer

- *public ResultMap answer(int questionId,String content)*

answer a short answer question

returns

ResultMap

parameters

-questionId:the id of the question

-content:the content of the answer

addResource

- *public ResultMap addResource(int questionId,int resourceId)*

add a resource to a question

returns

ResultMap

parameters

-questionId:the id of the question

-resourceId:the id of the resource

finish

- *public ResultMap finish(int taskId)*

help a worker to finish one task

returns

ResultMap

parameters

-taskId:the id of the taskId
