Programmable Web Project

Course Description

Spring 2025

521260S

5 ECTS

Staff: Mika Oja
Iván Sánchez

pwp-course@lists.oulu.fi

Table of contents

- Goals, learning outcomes and content.
- Course implementation
- Platforms
- Deadlines
- Evaluation
- Project work in Brief
- Miscellanea

Course goal

This course aims to provide adequate knowledge to design, implement, test and document a Web API.

Learning outcomes

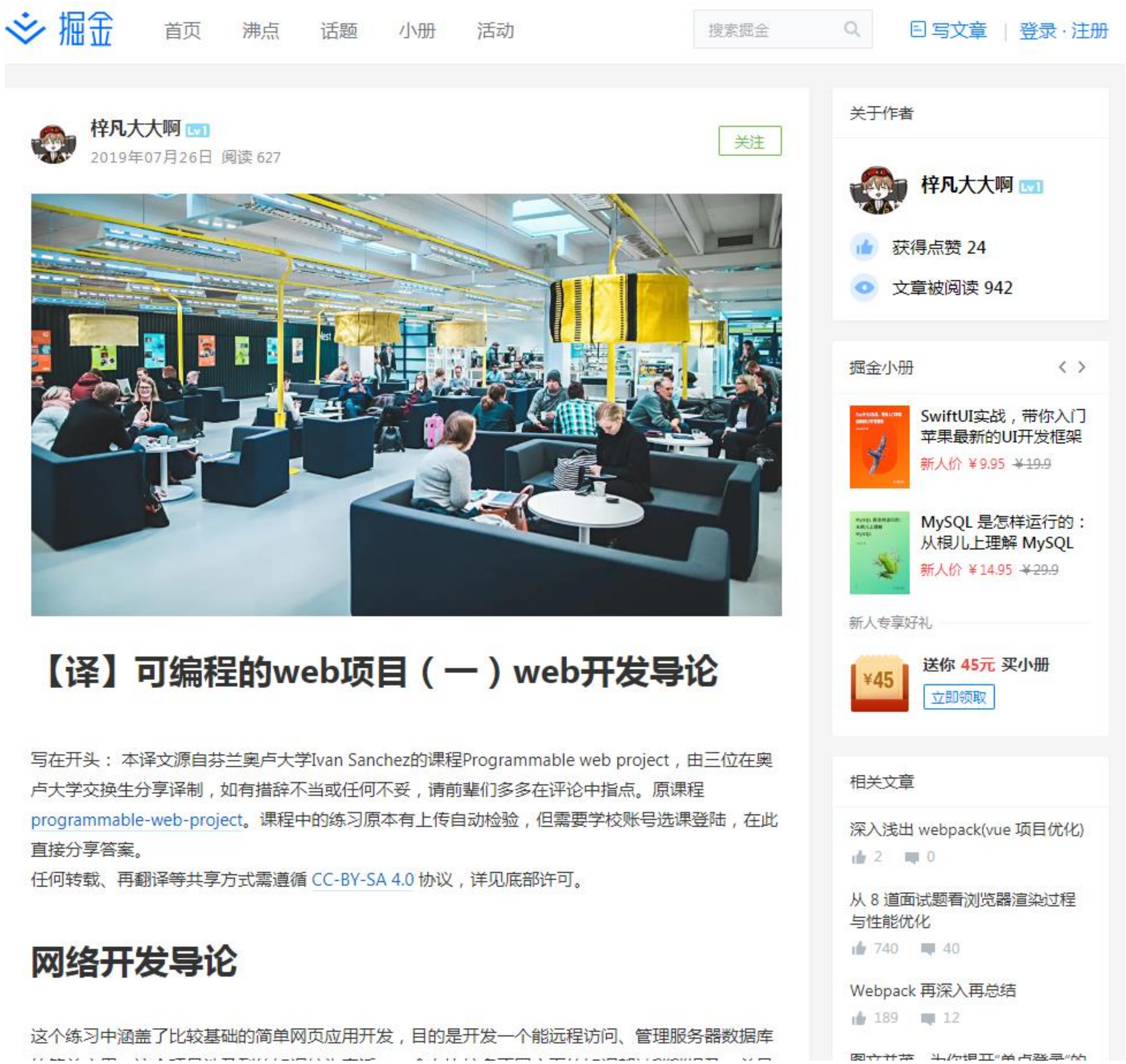
- Identify different Web API architectures and technologies (e.g. REST, RPC and event-driven architectures) being able to select the most appropriate for a system implementation.
- Understands Web API design principles and knows how to implement Web APIs using existing Web frameworks
- Evaluate Web API solutions being able to execute unit and functional testing for Web APIs
- Document Web APIs utilizing existing software tools (e.g. Swagger).
- Create clients or other services consuming the API.

Course content in Practice

- Introduction to concepts that are present in modern web development
  - (Relational) databases, APIs, web frameworks
- Understanding and designing interfaces for communication between services
  - Different types of APIs
  - REST APIs
  - Hypermedia
- Implementing web services and clients with Python
  - Using lightweight frameworks and libraries that do not hide too many details
  - Other technologies can be used in the course project as well

WHY THIS COURSE?

- **This course serves as an introductory course to API design and development**
  - Helps to develop BACKEND DEVELOPER SKILLS
- Full work cycle
  - Design, implementation, documentation and test
  - Several iterations based on customer (course staff) feedback
  - This work resembles quite a lot the way of working in IT
- Team work (3-4 people)
  - You need to define roles
  - You need to manage time.



梓凡大大啊 Lv1
2019年07月26日 阅读 627

【译】可编程的web项目（一）web开发导论

写在开头： 本译文源自芬兰奥卢大学Ivan Sanchez的课程Programmable web project，由三位在奥卢大学交换生分享译制，如有措辞不当或任何不妥，请前辈们多多在评论中指点。原课程programmable-web-project。课程中的练习原本有上传自动检验，但需要学校账号选课登陆，在此直接分享答案。
任何转载、再翻译等共享方式需遵循 CC-BY-SA 4.0 协议，详见底部许可。

网络开发导论

这个练习中涵盖了比较基础的简单网页应用开发，目的是开发一个能远程访问、管理服务器数据库

关于作者

梓凡大大啊 Lv1
获得点赞 24
文章被阅读 942

掘金小册

SwiftUI实战，带你入门苹果最新的UI开发框架
新人价 ¥9.95 ~~¥19.9~~

MySQL 是怎样运行的：从根儿上理解 MySQL
新人价 ¥14.95 ~~¥29.9~~

新人专享好礼

¥45 送你 45元 买小册
立即领取

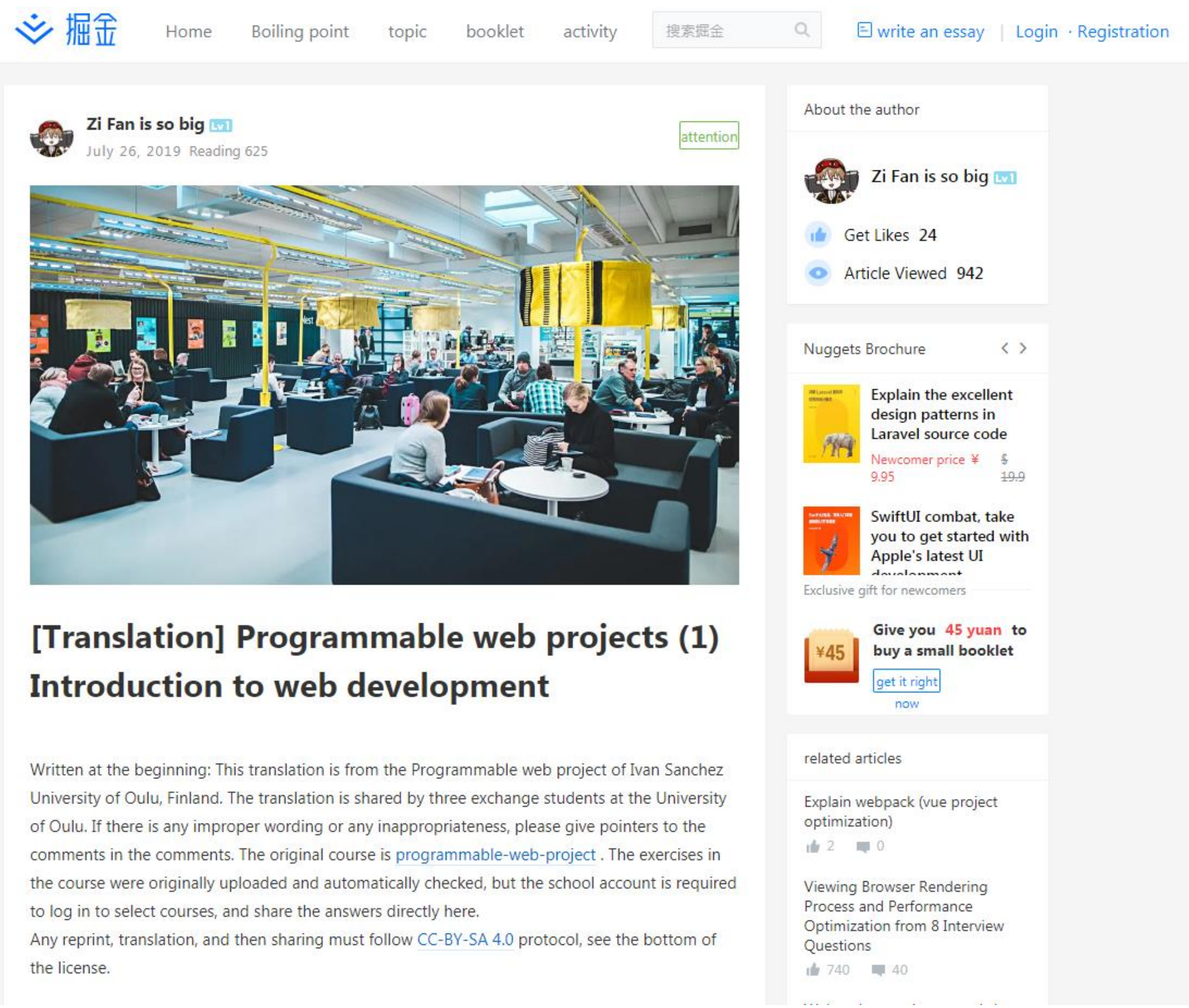
相关文章

深入浅出 webpack(vue 项目优化)
2 0

从 8 道面试题看浏览器渲染过程与性能优化
740 40

Webpack 再深入再总结
189 12

图文并茂，为你揭开“单点登录”的



Zi Fan is so big Lv1

July 26, 2019 Reading 625

[Translation] Programmable web projects (1) Introduction to web development

Written at the beginning: This translation is from the Programmable web project of Ivan Sanchez University of Oulu, Finland. The translation is shared by three exchange students at the University of Oulu. If there is any improper wording or any inappropriateness, please give pointers to the comments in the comments. The original course is programmable-web-project . The exercises in the course were originally uploaded and automatically checked, but the school account is required to log in to select courses, and share the answers directly here.

Any reprint, translation, and then sharing must follow CC-BY-SA 4.0 protocol, see the bottom of the license.

About the author

Zi Fan is so big Lv1

Get Likes 24

Article Viewed 942

related articles

Explain webpack (vue project optimization)

2 0

Viewing Browser Rendering Process and Performance Optimization from 8 Interview Questions

740 40

Course implementation

Practical approach

- **Project work**: Students design, document, implement and test a RESTful Web API.
  - Students implement small set of services
  - Interfaces between services will be documented
  - Different deadlines with intermediate feedback from course staff

- Students will work in GROUPS to practice the necessary communication and cooperation skills

Course implementation

Practical approach

- **Lecture** at the begining of the course provides the theoretical background
  - Lecture 1: Introduces Web Concepts
  - Lecture 2: Focus in APIs, hypermedia and introduces microservices

- **Exercises** provides necessary grodund work form implementing the project (workflow, toolchain, architecture...)
  - Each exercise is an online tutorial with small tasks
  - Each exercise deal with some aspect of the course project

Lecture

4 hours lecture (2+2 hours)

Lecture 1:

- Services and APIs
- Programmable Web
  - Definition and concepts
- Technologies for the Programmable Web
  - Databases
  - HTTP
  - Representation format: JSON, XML and Hypermedia
  - Web Clients

Lecture 2:

- Web APIs and Hypermedia
- Brief introduction to microservices

Exercises (I)

- 4 mandatory exercises
  - **Schedule in Tuudo / Peppi**

- Exercises instructions and tasks, as well as return box for the different tasks are available in Lovelace.

- Each exercise has two different parts:
  - Theoretical lecture (online???)
    - Link available in Lovelace
  - Q&A session (Computer rooms)
    - Course staff always present to answer student questions / providing tips …
    - You can do part of the work during the Q&A session BUT you won’t have time to complete the whole exercise in the given time.
    - Attending and participating actively in the Q&A sessions will provide extra points.

Exercises (II)

- Exercise 1: Introduction to Web Development.
- Exercise 2: Implemententig REST APIs with Flask.
- Exercise 3: API documentation and hypermedia.
- Exercise 4: Using an API (tentative)

Material and resources. Bibliography.

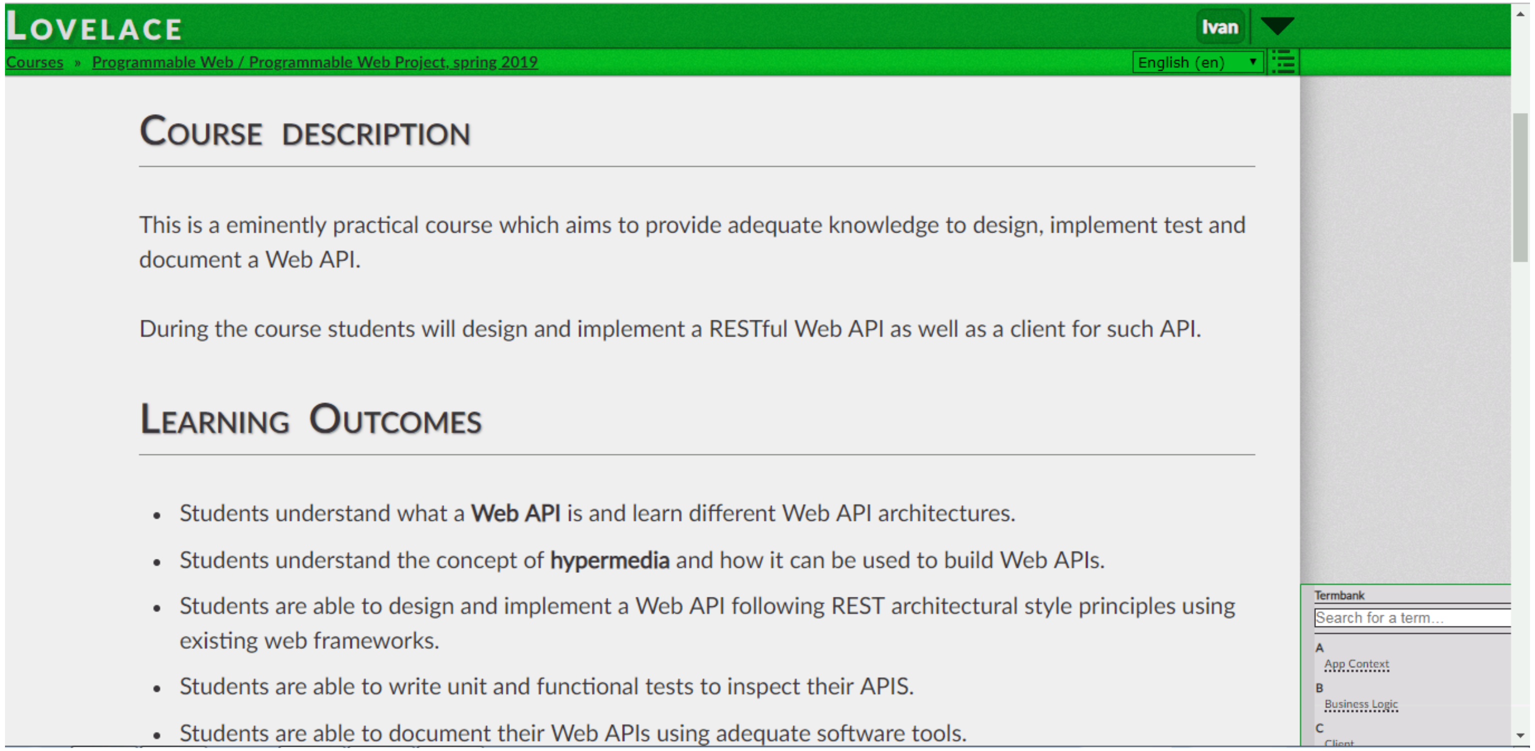
- Books:
  - **Leonard Richardson, Mike Amundsen, Sam Ruby. *RESTful Web APIs*. O'Reilly Media, 2013. ISBN: 978-1-4493-5806-8**
  - **Leonard Richardson & Sam Ruby,RESTful Web Services. O’Reilly Media 2007. ISBN: 978-0-596-52926-0.** Free available at http://restfulwebapis.org/rws.html

  An electronic version of the books are accessible through Oulu University Library catalogue.

- **Lecture and lab slides.**

- Extra study material will be provided during the course through Lovelace.

Platforms. LOVELACE



Course description

This is a eminently practical course which aims to provide adequate knowledge to design, implement test and document a Web API.

During the course students will design and implement a RESTful Web API as well as a client for such API.

Learning Outcomes

- Students understand what a **Web API** is and learn different Web API architectures.
- Students understand the concept of **hypermedia** and how it can be used to build Web APIs.
- Students are able to design and implement a Web API following REST architectural style principles using existing web frameworks.
- Students are able to write unit and functional tests to inspect their APIS.
- Students are able to document their Web APIs using adequate software tools.

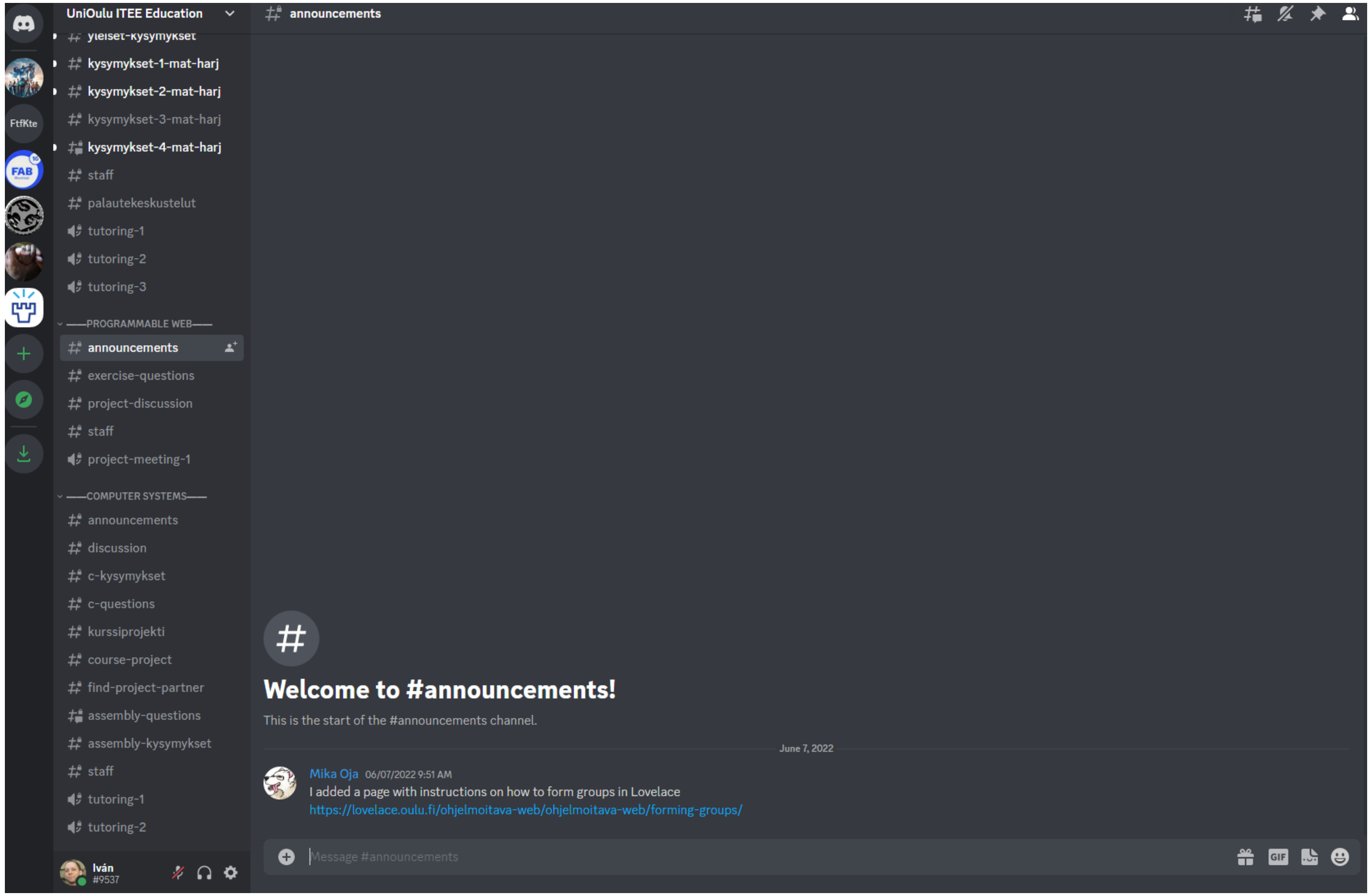
http://lovelace.oulu.fi

- Learning material
- Exercises instructions
- Return box for exercises and project deadlines
- Adminstrative tasks (group creations, meeting reservations…)

Platforms. LOVELACE

- Enrolling in Lovelace
- Creating teams
- Booking meeting times
- Returning exercises questions
  - Asking for help from course staff
- Returning project work

Platforms. Discord



Welcome to #announcements!

This is the start of the #announcements channel.

June 7, 2022

Mika Oja 06/07/2022 9:51 AM

I added a page with instructions on how to form groups in Lovelace
https://lovelace.oulu.fi/ohjelmoitava-web/ohjelmoitava-web/forming-groups/

- Announcements from course staff.
- “Instant” communication and feedback. Discussion

Platforms. Github / Gitlab

GitHub

https://github.com/

GitLab

https://about.gitlab.com/

- Project documentation
- Project code
- Meeting notes

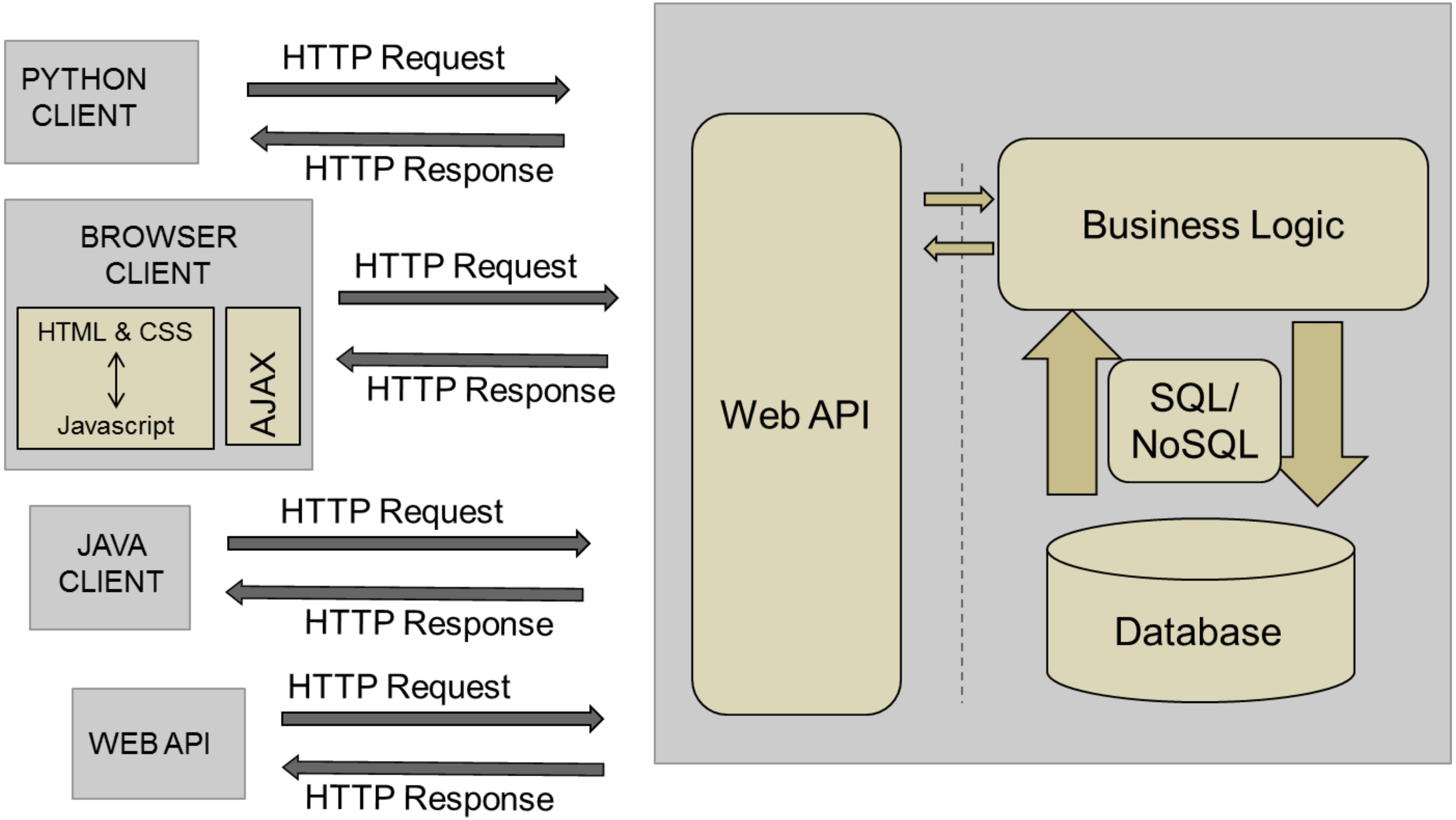
Platforms.Team communication

Project Work (I)

- In this project students design, implement, document and test:
  1. A functional **RESTful API**:
     - Meets REST principles, using the ROA architecture.
     - Utilizes Hypermedia or CRUD API to represent resources
     - Stores persistent data in a database (relational or not relational).
  2. **A Client application** that uses the Web API functionality:
     - It is composed at least by:
       - An HTTP client to access the RESTful API
       - A GUI / command tool (non-human driven clients accepted after contacting the course staff)
  3. An auxiliary service that communicates with the API and the client.
     - Performs operation that should be separated from API server
     - Does not need to be a REST service.

- More detailed requirements for the REST API, the client and the auxiliary service in Lovelace

Project Work (II)



Project Work (III)

- The project must be done in **groups of three / four people.**
  - **BE ACTIVE IN THE SEARCH OF A PARTNER**
  - Use Discord to find a partner. We will open a channel for that purpose.
    - Advertise your interest and the grade you are aiming for.

- One group member creates the group in Lovelace and invite other group members
  - If student created their own group cannot be invited to another group
  - Students cannot remove students from groups. Contact course staff.

Project Work in Brief (IV)

OPTION 1:

Deadlines

- The project is divided in **6 deadlines**
- Meeting with course staff after deadlines 1-4
- Missing one deadline -> automatically move to option 2

OPTION 2:

Final deliverable

- All the project content is delivered by the **final deadline**
  - **One intermediate meeting with staff is required**

Project work (V).

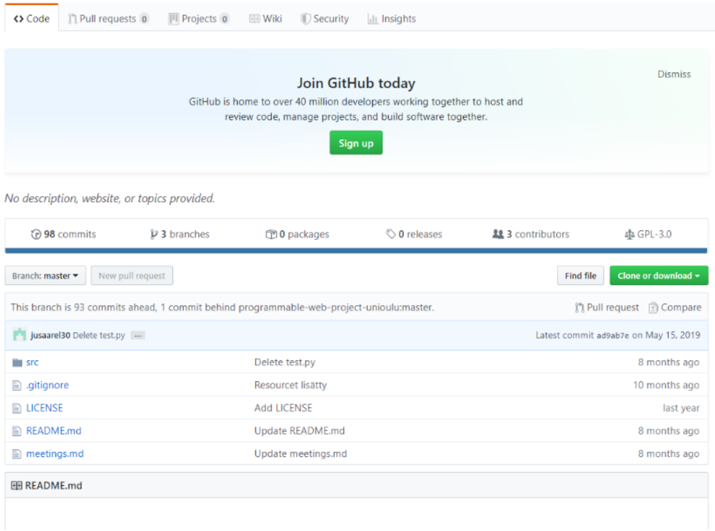
- You need to use a GIT repo to upload your code and documentation

GitHub: https://github.com/

GitLab: https://about.gitlab.com/

- Use the *GIT repo* for the code and the *Wiki* to document your project

- CLONE the GITHUB/GITLAB repo we provide. It contains the template for the documentation



Join GitHub today

GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.

No description, website, or topics provided.

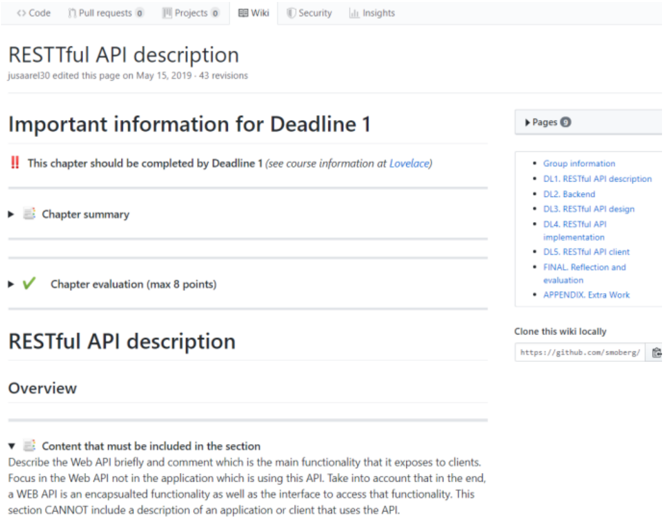
98 commits · 3 branches · 0 packages · 0 releases · 3 contributors · GPL-3.0

This branch is 93 commits ahead, 1 commit behind programmable-web-project-unioulu:master.

jusaarel30 Delete test.py … Latest commit ad9ab7e on May 15, 2019

src	Delete test.py	8 months ago
.gitignore	Resourcet lisätty	10 months ago
LICENSE	Add LICENSE	last year
README.md	Update README.md	8 months ago
meetings.md	Update meetings.md	8 months ago

README.md



RESTTful API description

jusaarel30 edited this page on May 15, 2019 · 43 revisions

Important information for Deadline 1

‼️ This chapter should be completed by Deadline 1 *(see course information at Lovelace)*

▸ Chapter summary

▸ ✓ Chapter evaluation (max 8 points)

RESTful API description

Overview

▾ Content that must be included in the section

Describe the Web API briefly and comment which is the main functionality that it exposes to clients. Focus in the Web API not in the application which is using this API. Take into account that in the end, a WEB API is an encapsualted functionality as well as the interface to access that functionality. This section CANNOT include a description of an application or client that uses the API.

- Group information
- DL1. RESTful API description
- DL2. Backend
- DL3. RESTful API design
- DL4. RESTful API implementation
- DL5. RESTful API client
- FINAL. Reflection and evaluation
- APPENDIX. Extra Work

Clone this wiki locally

https://github.com/smoberg/

Major Deadlines (Project)

- **Deadline 0** (19th Jan) : FOR ALL STUDENTS
  - Register the group and project topic in Lovelace

- **Deadline 1**: Project Plan(26.01.2025)
- **Deadline 2**: Database (09.02.2025)
- **Deadline 3**: API basic implementation(02.03.2025)
- **Deadline 4**: API documentation and hypermedia (30.03.2025)
- **Deadline 5**: API use / consumption(27.04.2025)
- **Deadline 6:** Project reflection and feedback(04.05.2025)

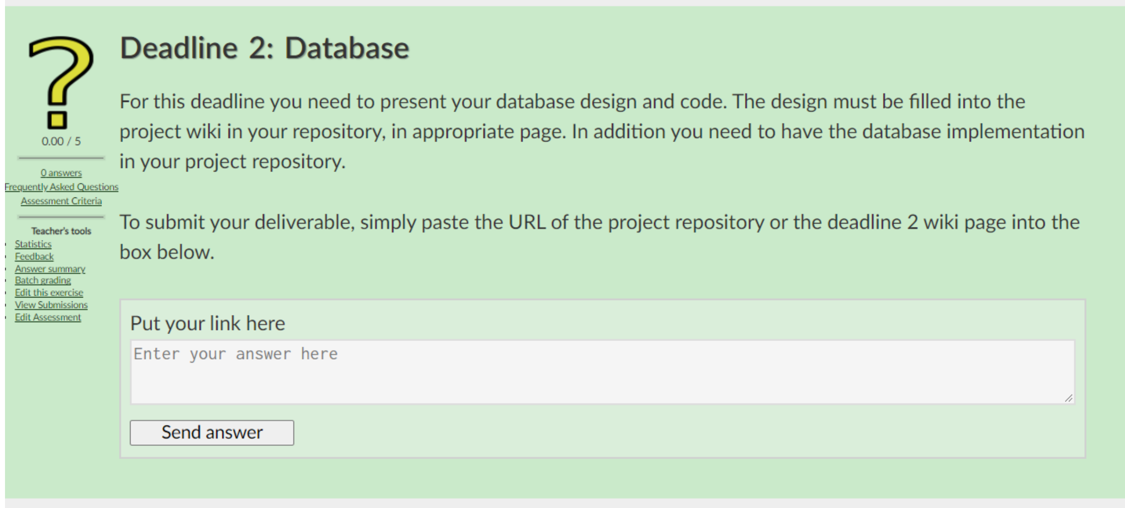
Students present the work done during the deadline in a meeting to the course staff.

NOTE: Experienced or otherwise confident groups who have trouble scheduling for the deadlines can do one final delivery instead.

- Students taking this option should have a midterm meeting with staff. Schedule will be assign later.

Project Work (VI)

- Before each deadline you must return in the corresponding Lovelace return box:
  - The link to your project wiki
  - The link to the repository



0.00 / 5

- 0 answers
- Frequently Asked Questions
- Assessment Criteria

Teacher's tools

- Statistics
- Feedback
- Answer summary
- Batch grading
- Edit this exercise
- View Submissions
- Edit Assessment

Deadline 2: Database

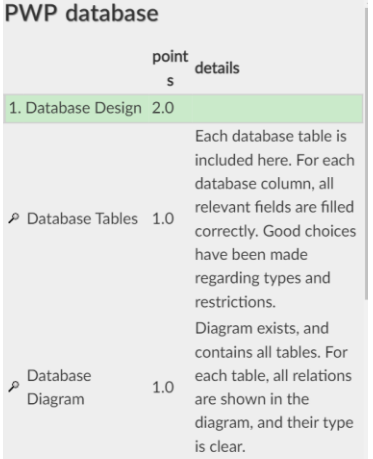
For this deadline you need to present your database design and code. The design must be filled into the project wiki in your repository, in appropriate page. In addition you need to have the database implementation in your project repository.

To submit your deliverable, simply paste the URL of the project repository or the deadline 2 wiki page into the box below.

Put your link here

Enter your answer here

- You can check the assessment criteria for each deadline in Lovelace



PWP database

	points	details
1. Database Design	2.0	
Database Tables	1.0	Each database table is included here. For each database column, all relevant fields are filled correctly. Good choices have been made regarding types and restrictions.
Database Diagram	1.0	Diagram exists, and contains all tables. For each table, all relations are shown in the diagram, and their type is clear.

Major Deadlines (Exercises)

Exercises deadlines will be Monday after the Q&A session.

- **Exercise 1.** TBD
- **Exercise 2.** TBD
- **Exercise 3.** TBD
- **Exercise 4.** TBD

- Delivering exercises late will have a penalization in the grade

- Attending to the Q&A for at least one hour will give you 1 extra point.

Assessment (I)

Project Work Topic	Deadlines	Points (out of 100) [*]
Project plan	D1	5
Database	D2	5
API Basic Implementation	D3	20
API Documentation and hypermedia	D4	15
API use (possible also deployment)	D5	30
Project reflection and feedback	D6	5
Project management and participation	-	4
Exercises	-	16

*** NOT DEFINITIVE**

- The final grade is obtained adding up the points of each deliverable.
  - Improving the deliverable by the final deadline => Increase the grade

- More accurate grading information will be published later in Lovelace

Assessment (II)

Grade 1::

- Complete first two exercises fully
- Implement a functional API server with basic features
- Document your API

Grade 2:

- Complete first three exercises fully
- Implement an API server with most graded features
- High quality code base and documentation

Grade 3:

- Complete all exercises
- Implement a functional API server and a functional client for it
- Document your API

Grade 4:

- Complete all exercises
- Implement an API server with most graded features
- Use hypermedia
- Make your client take full advantage of hypermedia
- Document your API

Grade 5:

- Same as 4, but also implement the third component