

Web Services & Service Oriented Architectures

Projects

2023-2024

Administrative

Registration

- Groups of 2 persons (binome)
- One of the project group's member has to fill in the form [L3 services web Project declaration 2023-2024.xlsx](#) to declare your project.
- **Deadline** to register your project and your group
Friday 16/2

Report and Submission

- It is obligatory to **use a distributed version-control system**, e.g., GitLab, GitHub, etc, **to collaborate with your partner while writing the code, and for the submission of the project.**
- By the end of the assignment one of the group's members has to **submit the link to your public project repository** including:
 - Your report (a README.md file) that should contain **at least**:
 - Member names and emails
 - Introduction describing the general purpose of the project, the different components, etc.
 - Services' description
 - Clients' description
 - At least one use case demonstrating the functionality of your implementation
 - The source code with **comments**
 - The WSDL of your service and Javadoc for your service code.
- **Submit the final link to your git in the corresponding section Project on the e-course platform**
- Deadline the 5th of April
- You will be convoked for a demonstration of your project!

Proposed Subjects

Requirements

- Web Services (JAVA)
 - **Develop** one web service with **at least** 2 operations.
 - Provide the web service as REST, and as WS-
 - Use JAX-WS and JAX-RS frameworks
 - Storing/retrieving the data in/from a database is a plus.
- Client (JAVA)
 - Call all the web service operations that you have implemented (REST and WS-*) from the same client.
 - Call also an **external service (WS-* or REST)**. (**Find many REST APIs here <https://rapidapi.com/hub>**).
 - **NOTE: You can call the external web service also from your own webservice.**
 - Combine the results of the calls with one another and with internal business logic.
 - Provide an interface for the users of your client (optional)
- Code
 - 50% reuse of online code is allowed (provide references)
 - Use comments
 - CHATGPT code generation is not allowed

Custom Projects

- You can propose your own project.
- It must fulfil the requirements defined before.
- It can have a thematic intersection with another project BUT TWO TEAMS CANNOT HAVE THE SAME PROJECT.
- Custom project descriptions must be validated by your professor, please contact me with the description of your project at aikaterini.tzompanaki@cyu.fr

Project1

E-shopping + Amazon/Asos/etc. Web service

Create an online purchase service, where users can:

- Get items belonging in a category of products
- Add/Remove product in a shopping list
- Get information (description, price) of a product

In the client, call also the external Web service to ask for the price of the same product on the e-platform. This can allow you to compute the difference in the price on your platform and a competitor platform.

Project2

Universities Management Service + Maps

Create service(s), which allow users to:

- Create a university, including its address.
- Create a specialty
- Connect specialties with universities

From the client, create universities, specialties and connect universities with specialties. For a given specialty, ask the names and addresses of the universities, and call the external service of google maps (<https://mapsplatform.google.com/> — free for up to 200\$ usage which should be enough for you) to show also the location of the universities (lng, lat).

Project3

Travel Agency + Weather Web service

- Create a service that adds a new country served by the agency. The service can also add specific cities (name, destination-type) served by the agency and connected to their countries. The service should provide the possibility to retrieve cities by destination-type.
- From the client add some cities (and countries) using the service operations. Then, call the webservice to find cities near a beach (using the destination-type) for a specific country. Moreover, show the weather at that city by calling an external service that gives the weather of the city (e.g., docs.stormglass.io).

Project4 (Beining YING, Jiaduo SUN)

Politicians+ Social media

- Create a service that provides the possibility to add and demand information of politicians and political parties.
- Create a client to add some politicians, and then ask for the information for a politician. Given their political party retrieved from the information returned, use a social media API, e.g., to retrieve the last posts/photos for the political party.

Project5

Music events + Music Web service

- Create a service that registers and provides information for upcoming events, either by area or by artist.
- From your client, use the web services to create some events and artists. Then, the client should display the events grouped by artist. Along with each event, include the nationality of the artist, retrieved from a music Web service, e.g., https://musicbrainz.org/doc/MusicBrainz_API, or developer.spotify.com/documentation/web-api, etc.

Project6 (Yige YANG)

Movie list + Movie Web service

- Create a service that
 - Adds movies
 - Adds actors
 - Links movies to actors
 - Retrieves information given a movie title,
 - Retrieves informations given an actor name.
- Create a client that
 - Consumes the aforementioned services
 - Interacts with the OMDb API service (www.omdbapi.com) to provide the ranking of the movie.

Project6-bis (Bruno POETA Maria AIT FERAHT)

Successful Movie list + Movie Web service

- Create a service that
 - Inserts to the local database a new movie
 - Updates the ranking of a movie, given its title.
 - Retrieves movies given a specific year span (from <date>.. to <date>)
 - Retrieves a movie given its local id.
- Create a client that
 - Consumes the aforementioned services. To insert a new movie, check first that the score of the movie is higher than 7 (only high-scored movies will be inserted).
 - To find the score of a movie, and the other movie information needed to add a new movie in the local database, call the OMDb api.

Project7

Lab members + Semantic Scholar

- Create a service that
 - Creates labs
 - Creates researchers
 - Links labs with researchers
 - Retrieves information of a lab, given its name
- Create client that
 - Consumes the aforementioned services
 - Interacts with the Semantic Scholar service (api.semanticscholar.org) to provide the average number of publications per person in a specific lab.

Project8 (Jianke LIN Hanchen LI)

Restaurant Management Service + Maps

Create service(s), which allow users to:

- Create a restaurant, including its address.
- Create a restaurant category (french, chinese, vegetarian, italian,..)
- Connect categories with restaurants

From the client, create restaurants, categories and connect restaurants with categories. For a given category, ask the names and addresses of the restaurants, and call the external service of google maps (<https://mapsplatform.google.com/> — free for up to 200\$ usage which should be enough for you) to show also the location of the restaurants (lng, lat).

Another possibility is to call the open street map api <https://www.openstreetmap.fr/donnees/>

Project9

Sports predictions site + FOOTBALL API

Create service(s), which allow users to:

- Create a team.
- Create a players
- Add/remove/update players to teams.

From the client, create teams and players and test the add/remove/update methods. Find the players for a given team, and then call an external API (eg , www.api-football.com/documentation) to find player statistics. Repeat for an opponent team and predict who would win in a potential match, based on the players' statistics.

Project10

Job hunting + Search

Create service(s), which allow users to:

- Add/remove/update a profile of a user.
- Retrieve job offers given a certain company.
- Match job offers with a specific user (match specialisation, localisation of job offers with users).

From the client, test the service for the user's management (create, update, delete). Then, test the job searching functionality of your service. To retrieve job offers, you can use the Jsearch API

<https://rapidapi.com/lets scrape-6bRBa3QguO5/api/jsearch/>

More...

- Instagram (developers.facebook.com/docs/instagram-api)
 - User's most popular posts
 - List of top user's fans
 - ...
- GitHub (developer.github.com)
 - Repos' stats
 - List of most used programming languages
 - ...
- Sports analytics (developer.boggio-analytics.com or www.api-football.com/documentation)
 - Fixtures
 - Team stats
 - Predictions
 - ...
- Finance (www.alphavantage.co/documentation)
 - Currency details
 - Compare prices
 - ...