

1 Requirements

1. You must have a working Ubuntu (or other Linux based) system installed **PRIOR** to this class. Failure to do so will prevent you from doing this assignment.
2. It is recommended, although not required, to work on this assignment in advance and to use the actual assignment day to ask questions and engage in discussion on the topic.

Contents

1	Requirements	1
2	Foreword	2
3	The HTTP protocol	2
3.1	Fields	2
3.2	Components	2
3.3	Experiments	2
4	REST APIs	2
4.1	REST paradigm	2
4.2	Exercises	2

2 Foreword

REST APIs are the standard to serve data and interact with a large number of systems. REST is built on the HTTP protocol and uses web servers to provide accessible APIs. First, we will see the standard http protocol and its structure, then we will look at how REST api leverage this protocol and how to build our own.

3 The HTTP protocol

Let's first have a look at the HTTP protocol.

3.1 Fields

1. What is the HTTP protocol original use?
2. What are the two parts of a HTTP transmission?
3. Find example of HTTP headers.
4. What are HTTP error codes?
5. What is the HTTP method field?
6. How are cookies stored?
7. What is the *Connexion* field used for?
8. How are Cookies used?
2. What are the different parts of a URL?
3. What is the different between an absolute and relative URL path?
4. What are URL parameters and how are they used?

3.2 Components

1. What is a URL?

3.3 Experiments

1. Use the python *http.server* package to serve a directory of your computer.
2. Print out different requests made to this server and look at the responses.
3. In python write a simple script that can query and save a web page.

4 REST APIs

Now let's dive in the REST paradigm and write our own REST APIs.

4.1 REST paradigm

1. For which reasons hve REST APIs become an industrial standard for building APIs?
2. What is the difference between a stateful and stateless API?
3. How are the different method values interpreted in the REST paradigm?
4. What is the path used for in REST APIs?
5. If you have multiple version of the same API that must exist at the same time how would you deploy them?
- on endpoint /
2. Implement different endpoints with different methods for a toy REST API of your choosing
3. Implement an endpoint to upload a file to your file
4. Implement a secured endpoint to you API that requires a simple user/password authentication with a hard coded user.
5. How would you implement an API that can return a large number of elements? Do it.

4.2 Exercises

1. Use the FastAPI python package to create a REST API displaying "Hello World" to a GET request
6. Connect your API to a MySQL database to save modifications made by users.