Faculty of Computers, Informatics and Microelectronics

Technical University of Moldova

PAD

Laboratory work # 7

REST.

Authors:
Petru Negrei
Victor Vasilica

Supervisor:
D. Ciorba

December 2014

1 Introduction

1.1 Topic

Describe the REST service.

2 Description

2.1 Definition

Representational state transfer (REST) is an abstraction of the architecture of the World Wide Web; more precisely, REST is an architectural style consisting of a coordinated set of architectural constraints applied to components, connectors, and data elements, within a distributed hypermedia system.
REST ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements.
The REST architectural style is also applied to the development of web services. One can characterize web services as "RESTful" if they conform to the constraints described in the architectural constraints section.

2.2 Content

During previous laboratory works we implemented a distributed system based on Restful principles and design the requests and responses based on that.

2.2.1 Structure

A uniform interface separates clients from servers. This separation of concerns means that, for example, clients are not concerned with data storage, which remains internal to each server, so that the portability of client code is improved. Servers are not concerned with the user interface or user state, so that servers can be simpler and more scalable. Servers and clients may also be replaced and developed independently, as long as the interface between them is not altered.

2.2.2 Stateless

The clientserver communication is further constrained by no client context being stored on the server between requests. Each request from any client contains all the information necessary to service the request, and session state is held in the client. The session state can be transferred by the server to another service such as a database to maintain a persistent state for a period and allow authentication. The client begins sending requests when it is ready to make the transition to a new state. While one or more requests are outstanding, the client is considered to be in transition. The representation of each application state contains links that may be used the next time the client chooses to initiate a new state-transition

2.2.3 Cacheable

As on the World Wide Web, clients can cache responses. Responses must therefore, implicitly or explicitly, define themselves as cacheable, or not, to prevent clients from reusing stale or inappropriate data in response to further requests. Well-managed caching partially or completely eliminates some clientserver interactions, further improving scalability and performance.

2.2.4 Uniform interface

The uniform interface constraint is fundamental to the design of any REST service. The uniform interface simplifies and decouples the architecture, which enables each part to evolve independently.

2.3 HTTP requests

HTTP based RESTful APIs are defined with these aspects. The following list shows the HTTP methods that are typically used to implement a RESTful API.
http://example.com/resources.

- **GET** - List the URIs and perhaps other details of the collection's members.
- **PUT** - Replace the entire collection with another collection.
- **POST** - Create a new entry in the collection. The new entry's URI is assigned automatically and is usually returned by the operation.
- **DELETE** - Delete the entire collection.

http://example.com/resources/item17.

- **GET** - Retrieve a representation of the addressed member of the collection, expressed in an appropriate Internet media type.
- **PUT** - Replace the addressed member of the collection, or if it doesn't exist, create it.
- **POST** - Not generally used. Treat the addressed member as a collection in its own right and create a new entry in it.
- **DELETE** - Delete the addressed member of the collection.

3 References

- Wikipedia `http://www.wikiwand.com/en/Representational_state_transfer`