

Phase 2

fbfreitas edited this page on Apr 10 · 7 revisions

Introduction

This document describes the requirements for the second phase of the Software Laboratory project.

Requirements

Command request structure

On phase 1, the command request structure was composed by three components: method, path and parameters.

```
{method} {path} {parameters}
```

For phase 2, this structure is added with a new *optional* component, called *headers*, placed after the path and before the parameters.

```
{method} {path} {headers} {parameters}
```

The headers component is composed by a sequence of name-value pairs, where each pair is separated by the '|' character. The name is separated from the value by the ':' character. An example is

```
accept:text/plain|file-name:movies.txt
```

Multiple visualisation formats

All the `GET` method commands must support the `accept` header, defining the format for the outputted representation:

- `text/plain` - plain text, similar to what was used in the first phase.
- `text/html` - Hypertext Markup Language (HTML).
- `application/json` - [Javascript Object Notation \(JSON\)](#).

An usage example is `GET /cinemas accept:text/html`

If the `accept` header is absent, then the `text/html` format should be used.

The `GET` methods must also support the `file-name` header, defining the file system location for the outputted representation. If this header is absent, then the representation is written into the standard output.

Interactive operation

If the application is called with empty arguments, then it should enter into an *interactive mode*. In this mode, the application repeatedly reads a line from the standard input and executes the corresponding command. This repetition is stopped by the execution of a command with `EXIT` method.

Ticket

A *ticket* is associated to one *session* and is characterized by:

- the row letter (i.e. from A to Z);

▼ Pages 12

[Home](#)

[calendar](#)

[Demonstration](#)

[discussion_calendar](#)

[environment](#)

[Introduction](#)

[Phase 0](#)

[Phase 1](#)

[Phase 2](#)

[Phase 3](#)


[Phase 4](#)

[turnos](#)

Clone this wiki locally

<https://github.com/isel-lei>



 Clone in Desktop

- the seat number in the row (i.e. a positive numeric value).

The application should support the following additional commands:

- `POST /cinemas/{cid}/theaters/{tid}/sessions/{sid}/tickets` - creates a new ticket given the following parameters:
 - `row` - row letter;
 - `seat` - seat number.
- `GET /cinemas/{cid}/theaters/{tid}/sessions/{sid}/tickets` - returns a list with all tickets for a session.
- `GET /cinemas/{cid}/theaters/{tid}/sessions/{sid}/tickets/{tkid}` - returns the detailed information of the ticket, including session information.
- `GET /cinemas/{cid}/theaters/{tid}/sessions/{sid}/tickets/available` - returns the number of available tickets for a session.
- `DELETE /cinemas/{cid}/theaters/{tid}/sessions/{sid}/tickets` - removes a ticket set, given the following parameter that can occur multiple times (e.g. `tkid=A3&tkid=B5`).
 - `tkid` - ticket identifier composed of the row letter and its number on the row.
- `GET /cinemas/{cid}/sessions/date/{dmy}` - return a list with the sessions in cinema `cid` in the date `dmy`, where `dmy` contains day, month and year.
- `GET /movies/{mid}/sessions/date/{dmy}` - returns a list with the sessions for the movie identified by `mid` in the day of the year `dmy` given one of the following optional parameters:
 - `city` - the city name;
 - `cid` - the cinema identifier;
 - `available` - the minimum number of available seats.

Additional commands

- `OPTION /` - presents a list of available commands and their characteristics.
- `EXIT /` - ends the application.

Delivery date

The phase 2 must be delivered until 21 of April, via the creation of a 0.2.0 tag on the GitHub repository.