



TECHNICAL
SPECIFICATION

2022

Mentors Nexus Application

WEB APPLICATION FOR CONTENT DELIVERY

Lubomír Rýgl

Mentors s.r.o.

Address

Contents

1	Version Control	5
2	Application Overview	6
2.1	short	6
3	Architecture Overview	7
3.1	General Architecture	7
3.2	AWS Architecture	7
3.3	Endpoint Overview	7
4	Endpoint Definitions	8
5	Database Design	9

Figures

Tables

1	Document version history	5
2	List of available endpoints	7

Listings

1	Backend Application base path	7
2	Format XML for MAC calculation example	9

1 Version Control

Version	Date	User	Change Note
ver1.0.0-3cbac7	14.03.2022	Lubomír Rýgl	Initial version

Table 1: Document version history

2 Application Overview

Mentors Nexus is a software web application used by Mentors.cz to provide its users access to educational materials online. The application back-end is running on Java 18 with SpringBoot Framework and the front-end will be a Angular application. For more detail about the availability and architecture - please see the documentation Architecture Overview. The following sections describe the actions that can be taken by users interfacing with the application.

2.1 Application

USER ACTION

- Register Account
- Log-In to Account
- Update User
- Delete User
- Subscribe to Course
- Purchase Course
- Watch Lesson
- Finish Lesson
- Finish Course

COURSE ACTION

- Create Course
- Edit Course
- Delete Course

LESSON ACTION

- Create Lesson
- Edit Lesson
- Delete Lesson
- Assign Lesson to Course

LEARNING PATH ACTION

- Create Learningn PATH
- Edit Learningn PATH
- Delete Learning Path
- Assign Course to Learning Path

3 Architecture Overview

3.1 General Architecture

3.2 AWS Architecture

Application shall be build as a three tier application leveraging AWS

3.3 Endpoint Overview

List of endpoints exposed to the frontend application available for request processing

```
1 https://mentors.cz/api/v01/
```

Listing 1: Backend Application base path

Allowed Method	Type	Endpoint	Description
GET	PUBLIC	/public-status	Shows application status
GET	PROTECTED	/status	Shows application status
GET	PUBLIC	/user/resetPassword	Reset user password
POST	PUBLIC	/user/register	Register a new user
POST	PUBLIC	/user/login	Log-In an existing user
POST	PROTECTED	/user/add	Add new user manually - requires a administrator permissions to perform this action. Endpoint is not publicly accessible
POST	PUBLIC	/user/register	Register a new user

Table 2: List of available endpoints

4 Endpoint Definitions

get	/api/v01/user/find/{id} <i>Get user record by user Id</i>
Parameter	
id	id of storage
Response	
application/json	
200	ok
<pre>1 { 2 "id": 867654678, 3 "name" : "Apfelmus", 4 "count" : 25 5 } 6</pre>	
404	error: storage not found
<pre>1 { 2 "message": "storage with id '11' not found!" 3 } 4</pre>	

post	/api/storage/ <i>create a new storage</i>
Parameter	
no parameter	
Body	
application/json	
<pre>1 { 2 "name" : "Apfelmus", 3 "count" : 25 4 } 5</pre>	
Response	
application/json	
200	ok
<pre>1 { 2 "id": 867654678, 3 "name" : "Apfelmus", 4 "count" : 25 5 } 6</pre>	

5 Database Design

```
1
2  /**
3   * Returns part of XML string from which MAC should be generated.
4   * param xml XML String
5   * param macElementName XML element name from which the MAC should be generated
6   */
7  fun getXmlPartForMacVerification(xml: String, macElementName: String): String {
8      val macStartElement = "<$macElementName>"
9      val macEndElement = "</$macElementName>"
10     if (xml.contains(macStartElement) && xml.contains(macEndElement)) {
11         val macData = xml.substring(
12             xml.indexOf(macStartElement),
13             xml.indexOf(macEndElement) + macElementName.length + TAG_OFFSET
14         )
15         val macString = macData.replace(">[\\s]*<".toRegex(), "><")
16         log.debug("MAC string length: {}", macString.length)
17         return macString
18     } else {
19         throw ParsingErrorException("Missing element $macElementName for MAC.")
20     }
21 }
```

Listing 2: Format XML for MAC calculation example