

CS393 - Project

Assignment Date : 6.12.2021

Deadline : 30.12.2021 09.00 AM

- This assessment must be prepared and finalized individually or as a team (3 people).
- If you are going to do the homework as a group, send me your group information within a week.
- You must upload your project into LMS as an archive file.
- I will **not** accept any meeting requests and will not answer questions after 6 pm the day before the deadline.

Suppose that you are working in a company as a software developer. Your company decided to develop a forum application like <https://stackoverflow.com>. A web application and a mobile application will be developed. You are included in the team that is responsible for developing server-side services. Your back-end application will provide services to these applications.

You should develop all required Restful services which are required for web/mobile applications. In the next page, you can find some screenshots from Stackoverflow. Parts enclosed in red box, are requested functionalities for both web and mobile applications.

Backend Functional Requirements:

1. You should create a Spring Boot web application. You should use JPA and Spring Data JPA. You are not allowed to use pure JDBC. But you can use native SQL queries in the Data JPA Repositories.
2. Develop all necessary Restful services to satisfy all the UI service requirements listed below.
3. Draw a class diagram for your model classes.
4. Generate an API documentation for your Restful services using OpenAPI (Swagger)
5. You are not allowed to return Entity Objects from Restful services. You must return custom DTOs. You should use the **mapstruct** library (<https://mapstruct.org/>) for such conversions.
6. Database selection is free of choice.
7. You should initialize your database automatically with at least one question (with title, text, tags, user) and two users. You are free to select any method you like for initialization.

UI Service Requirements (Frontend Functional Requirements)

1. Getting all the questions. User Interface (UI) only needs to display the question title, first 100 characters of question description, question tags, asked date, asked by, answer count, vote count. Optionally, the web application can send tags during the service call. If there are some tags in the request object, you should return all questions which have those tags. (Ex: getting all **springboot** related questions.)
2. Getting all information about a specific question for displaying question details on the screen.
3. Adding a new question. UI should send question title, question text, tags, and question owner to the backend. UI should get newly added question's id.
4. Adding a new answer for a question. In addition to the Question ID, UI should also send the following fields to the back-end application: answer text and user. UI needs newly added answer's id and related question's id.
5. Adding a new comment for a question. In addition to the Question ID, UI should also send the following fields to the back-end application: comment text and user. UI needs newly added comment's id and related question's id.
6. Adding a new comment for an answer. In addition to the Answer ID, UI should also send the following fields to the back-end application: comment text and user. UI needs newly added comment's id and related answer's id.
7. Voting a question, an answer, or a comment. UI needs to display updated vote count on the screen.
8. Deleting a specific comment(any types of comments)
9. Updating an answer
10. Updating a comment

CS393 - Project

Assignment Date : 6.12.2021

Deadline : 30.12.2021 09.00 AM

Grading

Providing a Class Diagram	5
Providing API documentation (UI)	5
Project architecture: <ul style="list-style-type: none">● Applying resource naming rules● Using 3-layer architecture: Presentation (Controller), Business Logic (Service), Data Access (Repository)	10
Restful Services	40
JPA, Repositories	30
Using mapstruct library for mapping entities to DTOs	5
Database initialization	5

CS393 - Project

Assignment Date : 6.12.2021

Deadline : 30.12.2021 09.00 AM

Listing questions

936
votes

How to configure port for a Spring Boot application

How do I configure the TCP/IP port listened on by a Spring Boot application, so it does not use the default port of 8080.

java spring spring-boot server port

asked Jan 13 '14 at 2:59

Paul Verest

54.1k 42 184 304

asked by

answer count

58
answers

882k views

435
votes

How to log SQL statements in Spring Boot?

I want to log SQL statements in a file. I have the following properties in application.properties
spring.datasource.url=... spring.datasource.username=user spring.datasource.password=1234 spring...

java spring hibernate spring-boot logging

asked May 8 '15 at 7:55

Oleg Pavliv

19k 7 56 72

435k views

419
votes

How to access a value defined in the application.properties file in Spring Boot

I want to access values provided in application.properties, e.g.: logging.level.org.springframework.web: DEBUG
logging.level.org.hibernate: ERROR logging.file=\${HOME}/application.log userBucket.path=...

java spring-boot properties-file

asked May 29 '15 at 11:19

Qasim

7,142 7 32 48

598k views

CS393 - Project

Assignment Date : 6.12.2021

Deadline : 30.12.2021 09.00 AM

Question Detail Page

Spring Boot: How can I set the logging level with application.pr

asked date
Asked 7 years, 11 months ago
Active 6 months ago
Viewed 573k times
view count

vote count
371

tags
logging spring-boot

Share Improve this question Follow

question title

question text
This is very simple question, but I cannot find information.
(Maybe my knowledge about Java frameworks is severely lacking)
How can I set the logging level with application.properties?
And logging file location, etc?

asked date
asked Dec 10 '13 at 2:43
zeodlr
9,031 13 38 57
asked by

comment text
comment writer
comment date

question comments
2 For the record, another option is to set the log level as environment variable, for example via the heroku dashboard. In Settings -> Config Vars: set logging.level.com.yourpackage to the desired level (INFO, ERROR, DEBUG). - LukeSolar Feb 25 '19 at 21:38
@LukeSolar I like that idea. Can you please explain how to read the value of environment variables and have it set under application.properties, such that it's different from development to testing (or production) environment? Or just redirect us to a link or something. I would appreciate that. - Kareem Jeiroudi Jul 17 '20 at 7:24
1 Hi @KareemJeiroudi - no need to read the value manually; you can use the pattern logging.level.my.package.name=DEBUG in properties file or define the variable in heroku with key 'logging.level.my.package.name' and value 'DEBUG' and that should set your logging level for the defined package. - LukeSolar Aug 18 '20 at 21:25

17 Answers
answer count

Active Oldest Votes

vote count
426

an answer
Update: Starting with Spring Boot v1.2.0.RELEASE, the settings in application.properties or application.yml do apply. See the Log Levels section of the reference guide.
logging.level.org.springframework.web: DEBUG
logging.level.org.hibernate: ERROR
For earlier versions of Spring Boot you cannot. You simply have to use the normal configuration for your logging framework (log4j, logback) for that. Add the appropriate config file (log4j.xml or logback.xml) to the src/main/resources directory and configure to your liking.
You can enable debug logging by specifying --debug when starting the application from the command-line.
Spring Boot provides also a nice starting point for logback to configure some defaults, coloring etc. the base.xml file which you can simply include in your logback.xml file. (This is also recommended from the default logback.xml in Spring Boot.
<include resource="org/springframework/boot/logging/logback/base.xml"/>
Share Improve this answer Follow
edited Jun 20 '20 at 11:58
answered Dec 10 '13 at 7:14
jarmann
445 6 14
M. Deinum
99.4k 20 192 198
answered by
answer date


answer comments
comment date
comment user
6 Normally anything that you can do on the command line works in an external config file. So debug=true would do it I think. That flag is a little bit special because the logging has to be initialized very early, but I think that would work. - Dave Syer Dec 10 '13 at 7:50
8 BTW the preferred logging system is logback, and that's what you get by default from the starter poms. So logback.xml is the most common tool to configure logging in a fine grained way. The --debug flag just switches on some selected Spring logging channels. - Dave Syer Dec 10 '13 at 7:54

CS393 - Project

Assignment Date : 6.12.2021








Deadline : 30.12.2021 09.00 AM

Sample Ask Question Screen

 Ask a question in Your Team

Title

Body

AA B I <>       

Markdown ☐

Try it yourself...

Tags
At least one tag is required.