
System Requirements Specification Index

For

REST API for Blog Application

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

1	Project Abstract	3
2	Assumptions, Dependencies, Risks / Constraints	3
2.1	Blog Constraints:	3
2.2	Comment Constraints	3
3	Business Validations	4
4	Rest Endpoints	4
4.1	BlogController	4

REST API for Blog APPLICATION

System Requirements Specification

1 PROJECT ABSTRACT

Blog Application is Spring boot RESTful application with MySQL, where it allows to manage the blogs and can post comments on the blog.

Following is the requirement specifications:

	Blog Application
Modules	
1	Blogs
2	Comments
Blog Module Functionalities	
1	Create a Blog
2	Update a Blog
3	Delete a Blog
4	Get the Blog by Id
Comment Module Functionalities	
1	Post a comment on the Blog

2 ASSUMPTIONS, DEPENDENCIES, RISKS / CONSTRAINTS

2.1 BLOG CONSTRAINTS:

- While fetching the Blog by Id, if Id does not exist then operation should throw custom exception.
- While Updating the Blog by Id, if Id does not exist then operation should throw custom exception
- While deleting the Blog by Id, if Id does not exist then operation should throw custom exception

2.2 COMMENT CONSTRAINTS

- If you want to post a comment on a blog, if blog id does not exists, then operation should throw a custom exception.

Common Constraints

- For all rest endpoints receiving @RequestBody, validation check must be done and must throw custom exception if data is invalid
- All the business validations must be implemented in dto classes only.
- All the database operations must be implemented on entity object only
- Do not change, add, remove any existing methods in service layer
- In Repository interfaces, custom methods can be added as per requirements.
- All RestEndpoint methods and Exception Handlers must return data wrapped in **ResponseEntity**

3 BUSINESS VALIDATIONS

- Blog title is not null, min 3 and max 100 characters.
- Blog content is not null, min 3, max 200 characters.
- Comment blog id is not null.
- Comment comment is not null, min 3 and max 200 characters.

4 REST ENDPOINTS

Rest End-points to be exposed in the controller along with method details for the same to be created

4.1 BOOKSCONTROLLER

URL Exposed		Purpose
1. /api/blogs		Successful Blog creation Response code: 201 (Created) If title or content is blank then Response code: 400 (Bad request)
Http Method	POST	
Parameter 1	BlogDto	
Return	BlogDto	
/api/blogs/{id}		If blog id is valid then Response code: 200 (OK) If blog id is invalid then Response code: 404 (Not Found)
Http Method	GET	
Parameter 1	Long(id)	
Return	BlogDto	
/api/blogs/{id}		For valid blog id Response code: 200 (OK) For invalid blog id: 404 (Not Found)
Http Method	PUT	
Parameter 1	Long(Id)	
Parameter 2	BlogDto	
Return	BlogDto	
/api/blogs/{id}		For valid blog id Response code: 200 (OK) For invalid blog id: 404 (Not Found)
Http Method	DELETE	
Parameter 1	Long(Id)	
Return	Boolean	

/api/blogs/comment		For valid blog id Response code: 200 (OK) For invalid blog id: 404 (Not Found)
Http Method	POST	
Parameter 1	CommentDto	
Return	CommentDto	