

# IMPACT.COM INTERVIEW PREPARATION PLAN

Interview: Friday, January 16, 2026 at 10:00 AM | 7 Days of Preparation

DAY	DATE	FOCUS	HOURS	KEY TOPICS
1	Fri 9	Java Fundamentals	6h	Java Core, Java 8 (Streams, Lambdas, Optional), JS Review
2	Sat 10	Spring Framework	7h	Spring Core (DI, IoC, Beans), Spring Boot, Vue.js Basics
3	Sun 11	REST APIs + React	8h	REST APIs, Spring Data JPA, React Hooks, System Design Intro
4	Mon 12	Security + Testing	8h	Spring Security, JWT, Unit Testing (JUnit, Mockito)
5	Tue 13	Kafka + Databases	8h	Apache Kafka, SQL Deep Dive, System Design Practice
6	Wed 14	DevOps + Mock	8h	Jenkins CI/CD, Docker, AWS, Mock Technical Interview
7	Thu 15	Final Review	6h	Review All Concepts, Behavioral Questions, Company Research

## DAILY SCHEDULE BLOCKS

DAY 1 - FRIDAY JAN 9	DAY 2 - SATURDAY JAN 10
Block 1 (2h): Java Core - OOP, Collections, Exceptions	Block 1 (2h): Spring Core - DI, IoC, Beans, Annotations
Block 2 (2h): Java 8 - Lambdas, Streams, Optional	Block 2 (2h): Spring Boot - REST API, Project Setup
Block 3 (1h): JavaScript ES6+ Review	Block 3 (2h): Vue.js - Data binding, Directives, Components
Block 4 (1h): CELPIP Practice	Block 4 (1h): System Design Intro
	Block 5 (1h): CELPIP Practice
DAY 3 - SUNDAY JAN 11	DAY 4 - MONDAY JAN 12
Block 1 (2h): REST APIs - CRUD, Annotations, ResponseEntity	Block 1 (1.5h): Spring Security Basics
Block 2 (2h): Spring Data JPA - Entities, Repositories	Block 2 (1.5h): JWT Authentication
Block 3 (2h): React - JSX, Components, useState, useEffect	Block 3 (2h): Unit Testing - JUnit, Mockito
Block 4 (1h): System Design - Microservices vs Monolith	Block 4 (2h): System Design - Load Balancing, Caching
Block 5 (1h): CELPIP Practice	Block 5 (1h): CELPIP Practice
DAY 5 - TUESDAY JAN 13	DAY 6 - WEDNESDAY JAN 14
Block 1 (2h): Apache Kafka - Producers, Consumers, Topics	Block 1 (1h): Jenkins CI/CD Pipelines
Block 2 (1h): Spring Kafka Integration	Block 2 (1h): Docker - Dockerfile, Containers
Block 3 (2h): SQL Deep Dive - JOINS, Indexes, Transactions	Block 3 (2h): AWS Basics - EC2, RDS, S3, Lambda
Block 4 (2h): System Design - Event-Driven Architecture	Block 4 (2h): Mock Technical Interview
Block 5 (1h): CELPIP Practice	Block 5 (1h): CELPIP Practice
DAY 7 - THURSDAY JAN 15	FRIDAY JAN 16 - INTERVIEW DAY
Block 1 (2h): Quick Review - All Key Concepts	8:00 AM: Wake up, breakfast, shower
Block 2 (2h): Behavioral Questions - STAR Method	9:00 AM: Warm-up review (1h)
Block 3 (1h): Company Research - impact.com	9:45 AM: Setup - camera, mic, water, paper
Block 4 (1h): Final System Design Review	10:00 AM: INTERVIEW - Smile, listen, be confident!
Evening: Prepare clothes, rest early (7-8h sleep)	Remember: You are prepared. You got this!

## KEY RESOURCES

TOPIC	RESOURCE	PRIORITY
Java 8 Features	<a href="https://www.youtube.com/watch?v=Q93JsQ8vcwY">youtube.com/watch?v=Q93JsQ8vcwY</a>	■■■
Spring Boot	<a href="https://www.youtube.com/watch?v=9SGDpanrc8U">youtube.com/watch?v=9SGDpanrc8U</a>	■■■
Spring Security	<a href="https://www.youtube.com/watch?v=b9O9NI-RJ3o">youtube.com/watch?v=b9O9NI-RJ3o</a>	■■
Kafka	<a href="https://www.youtube.com/watch?v=Ch5VhJza0al">youtube.com/watch?v=Ch5VhJza0al</a>	■■
Vue 3	<a href="https://www.youtube.com/watch?v=FXploQ_rT_c">youtube.com/watch?v=FXploQ_rT_c</a>	■■
React	<a href="https://www.youtube.com/watch?v=w7ejDZ8SWv8">youtube.com/watch?v=w7ejDZ8SWv8</a>	■■
Practice	<a href="https://start.spring.io">start.spring.io</a> + <a href="https://baeldung.com">baeldung.com</a>	■■■

## TOP 5 TECHNICAL QUESTIONS TO MASTER

#	QUESTION	KEY POINTS
1	What is Dependency Injection?	Pattern where dependencies are injected from outside, not created internally. Spring uses @Autowired.
2	Difference: @Component vs @Service vs @Repository	Single beans. @Service=business logic, @Repository=data access, @Component=generic.
3	What is Kafka and when to use it?	Message broker for event streaming. Use for: decoupling services, real-time processing, high throughput.
4	Explain REST API best practices	Use HTTP methods correctly, proper status codes, versioning, HATEOAS, stateless.
5	Microservices vs Monolith?	Monolith=one app, simple. Microservices=independent services, scalable, complex.

■ YOU ARE PREPARED. YOU GOT THIS. GOOD LUCK! ■