

01	☐ A	☐ B	☐ C	☐ D	☐ E
02	☐ A	☒ B	☐ C	☐ D	☐ E
03	☐ A	☐ B	☒ C	☐ D	☐ E
04	☐ A	☐ B	☒ C	☐ D	☐ E
05	☐ A	☐ B	☒ C	☐ D	☐ E
06	☐ A	☒ B	☐ C	☐ D	☐ E
07	☐ A	☐ B	☒ C	☐ D	☐ E
08	☐ A	☒ B	☐ C	☐ D	☐ E
09	☐ A	☐ B	☐ C	☒ D	☐ E
10	☐ A	☐ B	☒ C	☐ D	☐ E
11	☐ A	☐ B	☐ C	☐ D	☒ E
12	☐ A	☐ B	☐ C	☒ D	☐ E
13	☐ A	☐ B	☐ C	☐ D	☒ E

14	☒ A	☐ B	☐ C	☐ D	☐ E
15	☒ A	☐ B	☐ C	☐ D	☐ E
16	☒ A	☐ B	☐ C	☐ D	☐ E
17	☒ A	☐ B	☐ C	☐ D	☐ E
18	☒ A	☐ B	☐ C	☐ D	☐ E
19	☐ A	☐ B	☐ C	☐ D	☒ E
20	☐ A	☐ B	☐ C	☐ D	☒ E
21	☐ A	☐ B	☐ C	☒ D	☐ E
22	☐ A	☐ B	☐ C	☐ D	☒ E
23	☐ A	☒ B	☐ C	☐ D	☐ E
24	☐ A	☐ B	☐ C	☐ D	☒ E
25	☐ A	☐ B	☒ C	☐ D	☐ E
26	☐ A	☐ B	☒ C	☐ D	☐ E

27	☐ A	☐ B	☒ C	☐ D	☐ E
28	☐ A	☐ B	☒ C	☐ D	☐ E
29	☐ A	☐ B	☒ C	☐ D	☐ E
30	☐ A	☒ B	☐ C	☐ D	☐ E
31	☐ A	☐ B	☐ C	☒ D	☐ E
32	☐ A	☒ B	☐ C	☐ D	☐ E
33	☐ A	☒ B	☐ C	☐ D	☐ E
34	☐ A	☐ B	☒ C	☐ D	☐ E
35	☐ A	☒ B	☐ C	☐ D	☐ E
36	☐ A	☒ B	☐ C	☐ D	☐ E
37	☐ A	☐ B	☒ C	☐ D	☐ E
38	☐ A	☐ B	☐ C	☒ D	☐ E
39	☐ A	☐ B	☐ C	☐ D	☒ E

PRACTICE QUESTIONS

OutSystems

D011S

DevOps O11 (ES)

EDITION

Demo

QUESTIONS

9

ISSUED

September 3, 2026

Before you begin

What this is

9 practice questions for OutSystems **D011S**, each with its answer key and an explanation. Answer a question before you read its key — the explanation is worth more once you have committed.

If something looks wrong

Send us three things and we will check it:

- the exam code, **D011S**
- the question number
- the email address on your order

Write to **support@mometrix.net**. We reply within one business day.

Licence

This file is licensed to one person. Sharing or republishing it, in whole or in part, ends that licence.

QUESTION 1

SINGLE CHOICE

A DevOps engineer needs to deploy a previously tagged application version from the Development environment to Production using the OutSystems LifeTime API. Which endpoint should the engineer call first to determine the deployable version identifier?

- ☐ A POST /lifetime/rest/v1/deploy/plan
- ☒ B GET /lifetime/rest/v1/applications/{appName}/versions
- ☐ C POST /lifetime/rest/v1/deploy/versions/validate
- ☐ D DELETE /lifetime/rest/v1/deploy/versions/{versionId}

ANSWER B

WHY To deploy a specific tagged version via the LifeTime API, the engineer must first look up the application versions and capture the version identifier (the deploymentVersionKey). The GET applications versions endpoint returns the versions of an application along with their tags and identifiers. The deploy/plan endpoint is used to create a deployment plan but requires version keys that have already been resolved, and DELETE is unrelated to version discovery.

QUESTION 2

SINGLE CHOICE

Which LifeTime deployment step is mandatory before an application can be promoted from Quality to Production environment?

- ☐ A Running automated unit tests against the compiled application
- ☐ B Executing a successful deployment in the Staging environment
- ☐ C Obtaining business approval signatures in Service Center
- ☒ D Tagging the application version as 'Ready for Production'

ANSWER D

WHY OutSystems LifeTime enforces that only application versions tagged as 'Ready for Production' are eligible for promotion to the Production environment. Without this tag, the deployment plan cannot include the version when targeting Production. The other options describe valid DevOps practices but are not enforced as a prerequisite by the LifeTime deployment flow.

QUESTION 3 SINGLE CHOICE

When invoking the LifeTime API to deploy an application, which HTTP header must be included to authenticate service-to-service requests?

- ☐ A X-API-Key: <service-token>
- ☒ B Authorization: Basic <base64(user:password)>
- ☐ C Cookie: OS_LIFETIME_SESSION= <token>
- ☐ D Bearer <It-token> in the Authorization header

ANSWER B

WHY The LifeTime REST API uses HTTP Basic Authentication, where the credentials of a LifeTime administrator or a service account user are sent in the Authorization header as 'Basic <base64-encoded user:password>'. Personal Access Tokens and Bearer tokens are not part of the current LifeTime API authentication model. Cookies are used for the browser UI, not the REST API.

QUESTION 4 SINGLE CHOICE

An OutSystems developer is using the BDD Framework to create automated tests. Which component is responsible for executing the test scenarios defined in the application?

- ☐ A Service Center Test Runner
- ☐ B LifeTime Test Execution Engine
- ☒ C BDD Framework Test Runner module
- ☐ D Integration Studio Test Harness

ANSWER C

WHY The BDD Framework component installed in the OutSystems environment provides its own Test Runner that executes the test scenarios described using the Given/When/Then notation. Service Center does not host the BDD execution, LifeTime does not execute application-level BDD tests, and Integration Studio is unrelated to test execution in the web stack.

QUESTION 5 SINGLE CHOICE

Which of the following is a recommended OutSystems security best practice when exposing REST APIs to external consumers?

- ☐ A Disable authentication on the API to simplify integration
- ☐ B Expose internal database tables directly as REST endpoints
- ☒ C Apply a dedicated REST API role and use HTTPS with a valid certificate
- ☐ D Share the same service account credentials with all consumers

ANSWER C

WHY OutSystems security best practices recommend exposing REST APIs under a dedicated role with the minimum required permissions and serving them exclusively over HTTPS using a valid TLS certificate issued by a trusted CA. Disabling authentication, exposing internal tables, and sharing service accounts all violate the principle of least privilege and the platform's secure integration guidelines.

QUESTION 6 SINGLE CHOICE

Which OutSystems component provides centralized monitoring of application performance, errors, and infrastructure health across multiple environments?

- ☐ A Service Studio Debugger
- ☒ B LifeTime Console
- ☐ C Integration Studio Logger
- ☐ D OutSystems Now mobile app

ANSWER B

WHY LifeTime Console centralizes environment, deployment, and application monitoring information, providing visibility into deployed applications and environment health. Service Studio Debugger is a development-time tool, Integration Studio Logger relates to extension debugging, and OutSystems Now is a mobile consumption application, none of which provide centralized production monitoring.

QUESTION 7 SINGLE CHOICE

Which OutSystems tool enables version control, branch management, and conflict resolution when multiple developers work on the same OutSystems application?

- ☐ A Service Center
- ☐ B LifeTime Deployment Console
- ☐ C Integration Studio
- ☒ D OutSystems platform integrated with a Git repository

ANSWER D

WHY OutSystems provides native integration with a Git repository (configured in Service Center) to support version control, branching, merging, and conflict resolution of application modules. Service Center is the management console but relies on the configured Git provider for actual branching operations; LifeTime and Integration Studio do not perform collaborative version control.

QUESTION 8 SINGLE CHOICE

Which LifeTime concept allows a team to track every change applied to an application, including the source developer, timestamp, and target environment?

- ☐ A Application Tags
- ☒ B Deployment Plans and their audit history
- ☐ C Service Center eSpaces
- ☐ D Infrastructure components

ANSWER B

WHY Deployment Plans in LifeTime record every deployment operation, including the application versions involved, the environments, the user who triggered the action, and the timestamp. This audit history is the canonical mechanism for tracking change across environments. Application Tags label versions but do not record the audit trail, and Service Center eSpaces and infrastructure components describe structure rather than change history.

QUESTION 9

MULTIPLE CHOICE

Which of the following operations can be performed using the OutSystems LifeTime REST API as part of a CI/CD pipeline? (Choose two.)

- ☒ **A** Create and execute a deployment plan between environments
- ☐ **B** Compile native mobile applications for app store distribution
- ☒ **C** Register and manage application versions and their tags
- ☐ **D** Modify the visual theme of an application at runtime
- ☐ **E** Directly edit an OutSystems eSpace module's source code

ANSWER A - C

WHY The LifeTime REST API exposes operations to create and run deployment plans between environments, as well as to register and tag application versions. Native mobile build pipelines and visual theme runtime changes are handled by different tooling (e.g., Service Studio / Mobile Build Service), and source code modification is performed in Service Studio, not via LifeTime.