Boost up Your Certification Score

SAP
C_CPE
SAP Certified - Backend Developer - SAP Cloud Application Programming Model

Exams Boost
Boost Up Your Career

For More Information – Visit link below:

https://www.examsboost.com/

Product Version

✓ Up to Date products, reliable and verified.
✓ Questions and Answers in PDF Format.

Visit us at: https://www.examsboost.com/test/c-cpe

Latest Version: 4.1

1. Unified Scenario Simulation
2. Micro Skill Drill Exam

Topic: 1
Unified Scenario Simulation
Mussafah Precast: Building a Curing-Yard Extension on SAP BTP
Business Context
Mussafah Precast manufactures precast-concrete panels and structural components for commercial construction projects across Abu Dhabi, United Arab Emirates. The company runs SAP S/4HANA Cloud for its core manufacturing and procurement processes and follows a clean-core approach, keeping custom work outside the digital core. Its production planners have asked for a lightweight way to record each curing batch that panels pass through in the yard, together with the strength class each batch is expected to reach. Rather than change the core, the team agreed to deliver this capability as a side-by-side extension on SAP BTP built with the SAP Cloud Application Programming Model.
System Landscape
A backend developer set up the project in SAP Business Application Studio, using Core Data Services to model the data and planning to serve a SAP Fiori elements user interface for the yard supervisors. The extension is expected to read material and supplier details from the existing SAP S/4HANA Cloud system, so that each batch record shows the panel type and concrete supplier it relates to. The project is shared through a Git repository, and a CI/CD service is connected so the team's changes can move toward a deployable build.
Implementation Progress
Early in the build, the developer modeled an entity for curing batches, capturing the pour date, the yard bay, and the target strength class. A first preview of the Fiori elements interface opened without errors, and the navigation shell rendered as expected. When the supervisors opened the batch list, though, the page stayed empty, even though sample records had been added to the underlying table during testing.
Logic Context
The batch records also needed a computed ready-for-lift date, derived at the moment a batch is created from the pour date and the curing duration. The developer wrote handler code to calculate this value and registered it in the application's event flow. In a later check of the stored records, the pour date and yard bay were present, but the ready-for-lift date came back blank for every batch the supervisors had entered that morning.
Integration Context
The supplier and panel-type details from SAP S/4HANA Cloud were the next piece. The developer referenced the remote supplier entity in the service and expected the names to appear alongside each batch. During the walkthrough, the local batch fields displayed correctly, but the supplier column showed no values, and no connection to the remote system had yet been arranged for the extension.
Delivery Context
With a review approaching, the developer saved the latest changes in the development workspace and asked a colleague to check the pipeline output. The colleague reported that the CI/CD run still reflected the previous day's version, and none of the morning's handler changes were visible in the build. The team wants each of these observations resolved before the extension is promoted, while keeping the SAP S/4HANA Cloud core untouched.
Your assignment

Visit us at: https://www.examsboost.com/test/c-cpe

You are the backend developer for a side-by-side extension on SAP BTP built with the SAP Cloud Application Programming Model for Mussafah Precast. The extension records curing batches for precast panels and is expected to serve a SAP Fiori elements interface, compute a value at batch creation, display supplier details from SAP S/4HANA Cloud, and move through a Git-connected CI/CD pipeline. Work within SAP Business Application Studio and keep the SAP S/4HANA Cloud core unchanged. Complete the task expectations below and confirm each stated completion evidence

1. CHALLENGE 1 — Exposing the Curing-Batch Records as a Usable Service

• Confirm that the curing-batch entity defined in the data model is projected into an exposed service so it is reachable over OData.
• Bind the SAP Fiori elements list to that exposed service, not to the persistence entity alone.
• Distinguish a rendered navigation shell from a service that actually returns batch records.

Checkpoints

• Querying the exposed service endpoint returns the test batch records.
• The supervisors' batch list displays the stored records after the interface is bound to the exposed service.

2. CHALLENGE 2 — Stamping the Ready-for-Lift Date at Batch Creation

• Ensure the ready-for-lift date is computed from the pour date and curing duration when a batch is created.
• Register the computation so it runs before the record is persisted, so the value is stored with the batch.
• Keep the computed value inside the extension rather than adding it to the core material record.

Checkpoints

• A newly created batch shows a populated ready-for-lift date in the stored record, not only on screen.
• The SAP S/4HANA Cloud core carries no new custom field for this value.

Question: 1

CHALLENGE 1 — Exposing the Curing-Batch Records as a Usable Service

The supervisors' batch list stays empty even though test records exist in the underlying table, while the navigation shell renders cleanly.
What is the most likely reason the batch records are not appearing in the interface?

A. The Fiori elements shell rendered without errors, so the interface is complete and the sample records were likely inserted into the wrong table during testing.
B. The preview page opened cleanly, so the service is already working and the supervisors probably lack the reading role that the batch list requires.
C. The batch entity was modeled for storage but never projected into an exposed service, so no OData endpoint returns those records to the interface.
D. The navigation shell loaded correctly, so the data model is finished and the records will surface once the application is redeployed to its runtime.

Answer: C

Explanation:

Visit us at: https://www.examsboost.com/test/c-cpe

In CAP the persistence entity and the exposed service are separate artifacts, and an entity is not reachable over OData until a service projects it. A rendered shell only proves the interface loaded, not that any service returns rows, which is why the list is empty despite records existing.

Question: 2

CHALLENGE 1 — Exposing the Curing-Batch Records as a Usable Service

After the batch entity is projected into a service, the developer wants to confirm the records are now reachable by the interface.
Which check best confirms that the batch records are now served over OData?

A. Query the exposed service endpoint for the batch entity and confirm the test records are returned before opening the Fiori elements list.
B. Reopen the navigation shell and confirm it still renders, treating a clean shell load as evidence that the records are now being served.
C. Re-insert the sample records into the table a second time and assume the list will populate because the batch data now exists twice over.
D. Redeploy the whole application to its runtime first and treat a successful deployment as sufficient confirmation that the entity is now reachable.

Answer: A

Explanation:
Querying the exposed service endpoint directly shows whether the entity now returns rows over OData, which is the precise evidence the interface depends on. It isolates the service layer from the UI, confirming the fix before the list is opened.

Question: 3

CHALLENGE 1 — Exposing the Curing-Batch Records as a Usable Service

The batch entity is modeled, and the developer needs to take it to a working list in the Fiori elements interface.
Which ordered set of steps correctly makes the modeled entity appear as a working list in the interface?

A. Bind the interface to the entity, then project the entity into the service, then confirm the batch records are returned.
B. Redeploy the application, then project the entity into the service, then bind the interface to the entity afterwards.
C. Project the entity into the service, then redeploy the whole application, then insert additional sample records into the table.
D. Project the entity into the service, then bind the interface to the exposed service, then confirm the batch records are returned.

Answer: D

Explanation:
The entity must first be projected into a service so an OData endpoint exists, then the interface must be bound to that exposed service, and finally the records are confirmed. This order respects the real dependency: nothing can be bound or returned before the service exposes the entity.

Question: 4

CHALLENGE 2 — Stamping the Ready-for-Lift Date at Batch Creation

The ready-for-lift date is blank on every stored batch even though the handler code runs when a batch is created.
What is the most likely cause of the blank stored value?

A. The computed value is calculated correctly but the interface hides the column, so the date already exists in storage and only the display needs adjusting.
B. The handler was registered to run after the record is persisted, so it computes the date too late to be written into the batch that was just saved.
C. The pour date is missing from the data model, so the calculation has no input, and the ready-for-lift date should instead be entered manually by supervisors.
D. The curing duration lives in the core system, so the extension cannot read it, and the ready-for-lift date must be maintained directly in SAP S/4HANA Cloud.

Answer: B

Explanation:
Logic that must influence a stored value has to run before the record is persisted; a handler registered to run afterward sees the already-saved record and cannot change what was written. That timing is exactly why the code executes yet the stored date stays blank.

Question: 5

CHALLENGE 2 — Stamping the Ready-for-Lift Date at Batch Creation

The developer needs the ready-for-lift date to be stored with each new batch, not just shown at view time.
What is the most appropriate way to ensure the value is stored with each new batch?

A. Recalculate the date whenever a batch is read, accepting that stored records stay blank while the interface shows a value only at view time.
B. Add a scheduled job that fills the blank dates each night, leaving every new batch without the value until the next overnight run completes.
C. Ask supervisors to type the ready-for-lift date manually at creation, moving the calculation out of the handler and onto the yard team instead.
D. Register the calculation in the before-persistence phase of the event flow, so the value is set on the record before it is saved.

Visit us at: https://www.examsboost.com/test/c-cpe

Answer: D

Explanation:
Running the calculation in the before-persistence phase sets the value on the record before it is written, so it is stored with every new batch. This addresses the timing directly rather than compensating for it elsewhere.

Topic: 2
Micro Skill Drill Exam

Question: 6

A developer at Klöber Medizintechnik, a German medical-device manufacturer in Stuttgart, secures a CAP-based calibration-tracking extension on SAP BTP. The access design states that only users assigned a Calibration Auditor role may read calibration-audit entries, while every authenticated user may read the general equipment list. The developer enabled authentication so unauthenticated calls are rejected, and the equipment list now works for each signed-in user. However, any authenticated user — including those who do not hold the auditor role — can also read the calibration-audit entries, which breaks the access design. During a quick check, a tester holding only a basic viewer profile opened the audit list and saw calibration timestamps and technician identifiers in full. A colleague points out that requiring a signed-in user proves identity but does not by itself limit an operation to a particular role, and that no role-based restriction was ever placed on the audit entity.
What is the correct way to enforce that only users with the Calibration Auditor role can read the calibration-audit entries?

A. Assign the Calibration Auditor role to every authenticated user in the identity service, so the role check passes consistently across the whole application.
B. Remove the authentication requirement from the equipment-list service, so a single stricter default applies everywhere and blocks non-auditors from the entries.
C. Move the calibration-audit entity into a separate database schema, since isolating the data at the storage layer keeps non-auditor users from reading it.
D. Add a role restriction on the calibration-audit entity that limits its read event to the Calibration Auditor role, so only that role is authorized.

Answer: D

Explanation:
Authentication proves identity, but limiting an operation to a role requires an access restriction on the entity that scopes the read event to that role. Adding a restriction for the Calibration Auditor role is what stops authenticated non-auditor users from reading the audit entries, matching the access design.

Question: 7

A developer at Naniwa Retail, an Osaka-based store chain, builds a CAP extension that must read product master data from an existing SAP S/4HANA Cloud system and combine it with locally stored promotion data. The developer wrote the projection and the user interface expecting product fields to

appear, but the product data never loads, while the local promotion data displays correctly. On review, the project contains no imported definition of the external S/4HANA service, and nothing in the configuration tells the CAP application where the remote system is or how to reach it. The developer had assumed that simply referencing the product entity in the projection code was enough to pull the data across from S/4HANA. On screen, the promotion tiles render with their prices and validity dates, yet every product name and description in the combined view stays blank, and the application log shows no outbound call ever leaving for the remote system.
What must be in place for the CAP application to retrieve the product master data from the external SAP S/4HANA service?

A. Recreate the product projection using the same field names as S/4HANA, since matching names let the framework resolve and load the remote data automatically.
B. Import the external service's definition into the project and configure a connection to the remote S/4HANA system, so the app can call it for product data.
C. Add the product fields as new columns on the local promotion entity, so the missing product data is served from the local database next to the promotions.
D. Increase the page size on the product projection, because the product rows are being retrieved from S/4HANA but truncated before they reach the user interface.

Answer: B

Explanation:
Consuming a remote service needs both the service's definition imported into the project and configured connection details for the remote system. Without the contract and a destination, the application has nothing to call, which is why the product data never loads; supplying both lets the app retrieve it from S/4HANA.

Question: 8

A developer at Bow River Utilities in Calgary, Canada, is ready to deploy a completed CAP extension to the SAP BTP Cloud Foundry runtime for the first time. The developer built the deployable application archive and then ran the deployment command, but the runtime rejected it before any upload began, reporting that no target environment was set for the session. The archive itself is valid and the application ran cleanly in local testing. A colleague explains that a Cloud Foundry deployment is only accepted inside a session that has been authenticated and pointed at a specific organization and space, and that this targeting step was skipped. The developer had signed in to the platform earlier in the week, but the current terminal session carried no active login and no selected space when the deploy command was issued. The team wants to know which single prerequisite was missing so the first deployment can proceed.
Which missing prerequisite step must be completed so the Cloud Foundry deployment is accepted?

A. Log in to Cloud Foundry and target the correct org and space before deploying, since the runtime rejects a deploy issued from an untargeted session.
B. Scale the application to additional instances before deploying, since the runtime will not accept an initial deploy for an app that runs on a single instance.
C. Enable the application autoscaler before deploying, since the runtime blocks any deployment for an application that has no scaling policy defined for its workload.

D. Bind a database service instance to the application before deploying, since the runtime rejects the deploy until every service dependency is already in place.

Answer: A

Explanation:
A Cloud Foundry deployment is accepted only within an authenticated session that targets a specific org and space. The developer deployed from a session with no target, which is exactly the rejection reported, so logging in and selecting the target is the missing upstream prerequisite.

Question: 9

At Cerrado Grãos, a Brazilian agribusiness cooperative in São Paulo, a developer extends a produce-grading service built with the SAP Cloud Application Programming Model. A requirement states that whenever a new grading record is created, the system must automatically stamp a calculated quality score onto the record before it is stored. The developer added the calculation inside a handler registered to run after the create operation completes. During testing the grading records are persisted correctly, but the quality-score field is always empty in the saved data, even though the calculation code runs without any error. A reviewer notes that logic which must influence the values written to the database has to execute at the right point in the request's event flow, and that work performed once persistence has already happened cannot change what was saved for that operation.
How should the custom logic be registered so the calculated quality score is stored with each new record?

A. Keep the after-create handler and add a read handler that recomputes the score, so the value is filled in whenever the record is later retrieved for display.
B. Move the calculation to a handler that runs before the create operation persists the record, so the computed quality score is part of the saved data.
C. Keep the after-create handler but also grant the service user write authorization on the score field, since the empty value reflects a missing permission.
D. Register the calculation as a start-up handler that runs when the service boots, so every new record inherits the score from the initialized service state.

Answer: B

Explanation:
Values that must be persisted have to be computed before the record is written. A handler that runs before the create operation persists the row can set the quality score so it becomes part of the stored data, which is exactly what the requirement asks and what the current after-create timing prevents.

Question: 10

A developer at Table Bay Financial, a Cape Town financial-services firm, works on a CAP extension in SAP Business Application Studio. The team's practice is that changes reach the deployed test environment only through an automated CI/CD pipeline connected to the shared Git repository; the pipeline runs when a new change lands in that repository. The developer finished a change, confirmed it worked in

the local preview, and told the team it was ready to test. However, the reviewers cannot see the change and the pipeline never ran. Checking the project, the developer had saved the files in the Business Application Studio dev space but had not committed and pushed the change to the shared Git repository that the pipeline watches, so nothing new ever arrived for it to act on.
What must the developer do so the change enters the automated build-and-deploy pipeline?

A. Commit the change and push it to the shared Git repository the pipeline watches, so the new commit triggers the pipeline to build and deploy it.
B. Increase the pipeline's scheduled build frequency, so it eventually picks up the locally saved change without any new commit to the shared repository.
C. Re-run the local preview once more, since a successful local run is the signal the pipeline waits for before it builds and deploys the finished change.
D. Restart the Business Application Studio dev space, so the saved local files are re-read and forwarded into the build-and-deploy pipeline automatically.

Answer: A

Explanation:
The pipeline is triggered by a change arriving in the shared Git repository, and local edits saved only in the dev space are invisible to it. Committing and pushing publishes the change to the watched repository, which is what starts the build-and-deploy run.

Visit us at: https://www.examsboost.com/test/c-cpe

Thank You for Trying Our Product

For More Information – Visit link below:

https://www.examsboost.com/

15 USD Discount Coupon Code:

G74JA8UF

FEATURES

- ✓ 90 Days Free Updates
- ✓ Money Back Pass Guarantee
- ✓ Instant Download or Email Attachment
- ✓ 24/7 Live Chat Support
- ✓ PDF file could be used at any Platform
- ✓ 50,000 Happy Customer