Boost up Your Certification Score

SAP
C_CPI
SAP Certified - Integration Developer

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-cpi

Latest Version: 4.1

1. Unified Scenario Simulation
2. Micro Skill Drill Exam

Topic: 1
Unified Scenario Simulation
Vistula Mutual Health Opens Claims Status to Partner Clinics
Business Context
Vistula Mutual Health administers reimbursement plans for members across southern Poland from its offices in Kraków, and works with a network of independent partner clinics that submit member claims for treatment. Membership has grown after the insurer added two new regional plans, and the clinics now submit several thousand claims each month. Clinic staff currently phone the insurer's service desk to ask whether a submitted claim has been received, is under review, or has been settled, and those calls have become a daily burden on both sides. To ease that load, leadership approved a small initiative that lets each clinic's own booking application check a claim's status directly.
Platform Landscape
The insurer runs its claims processing in an SAP system and has recently adopted SAP Integration Suite on SAP Business Technology Platform to connect that system to the outside world. A single integration developer, guided by an occasional mentor, owns the work. Using the API Management capability, the developer built an API proxy that returns a claim's current status when it is called, and confirmed in testing that the proxy returns the right status for sample claim numbers. The intent is for each partner clinic's app developers to consume that API from their own applications rather than telephoning the service desk.
Release Objective
The goal for the first release is modest: a clinic application should be able to look up a single claim's status on demand and show it to reception staff immediately. The developer finished the proxy, tested it privately, and told the pilot clinics that the claims-status service was ready for them to connect. A few clinic developers were invited to begin building against it that week, and the team turned its attention to onboarding the next group of clinics.
Early Access Observations
Several days later, the pilot clinic developers reported that they could not find the claims-status service anywhere in the Developer Hub, so they had nothing to subscribe to and could not begin building. The mentor recalled that the proxy had been created and tested, but no one could point to a later step that would have offered it to app developers through the Hub. Separately, once a first clinic was finally connected and subscribed, its early calls were being refused, and a teammate assumed the offering simply needed to be published again to clear the problem.
Access and Volume Considerations
Attention then turned to how the service should behave once many clinics call it at once. Compliance asked that every calling application be verified and that no single clinic be able to query far more often than its share, so that member data is not exposed broadly and the claims system is not overwhelmed at peak. One developer suggested instead building a nightly routine that copies every claim's status into a shared file the clinics could read, arguing it would be simpler than exposing a live service. The team weighed the appeal of that batch approach against the plain requirement for an immediate, per-claim answer.
Governance Context

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

Finance and compliance also asked where the new access rules should live. One suggestion was to add the caller checks and the query limits inside the claims system itself, which felt direct. The mentor pushed back, pointing to the insurer's stated aim to keep the SAP claims core standard so that future updates stay easy to adopt. With the first clinics waiting and more plans launching soon, the team wants the lookup live quickly while still leaving room to admit many more clinics later without reworking the claims system each time.

Your assignment

You are acting as the integration developer for Vistula Mutual Health. A claims-status API proxy has been built in the API Management capability of SAP Integration Suite so that partner clinic applications can look up a single claim's status on demand, and pilot clinic developers are waiting to connect. Work in the shared development tenant. Where an exact technical setting is not specified, apply the general behavior of the platform rather than assuming a specific named option. Your task is to make the API discoverable to clinic developers, choose the capability that fits an on-demand lookup, place access controls without changing the claims system, and correctly separate discoverability from runtime access

1. CHALLENGE 1 — Publishing the Claims-Status API So Clinic Developers Can Find It

• Review the current state of the claims-status API, noting that the proxy works in private testing but that clinic developers cannot find anything to subscribe to in the Developer Hub.

• Determine the step required to offer the working API to app developers, so it becomes discoverable and subscribable through the Developer Hub.

• Avoid rebuilding a proxy that is already complete; act on the existing artifact.

Checkpoints

• A pilot clinic developer can locate the product that bundles the API in the Developer Hub.

• The same developer can complete a subscription, confirming the service is available end to end before the rest of the clinics are invited.

2. CHALLENGE 2 — Selecting the Right Integration Suite Capability for On-Demand Lookups

• Confirm the requirement that a clinic application receive an immediate status answer for a single claim on demand.

• Decide which SAP Integration Suite capability fits that request-and-response interaction rather than a bulk copy or a broadcast of every change.

• Weigh the simplicity of a scheduled copy against the need for a live answer and the requirement to share only what each lookup needs.

Checkpoints

• A single lookup returns a live status for one claim without distributing the full set of claim statuses.

• The chosen capability is justified by the on-demand interaction, not by which option felt simplest to build.

Question: 1

CHALLENGE 1 — Publishing the Claims-Status API So Clinic Developers Can Find It

The integration developer built and privately tested the claims-status API proxy, yet the pilot clinic developers report they cannot find anything to subscribe to in the Developer Hub.
What is the most likely reason the clinic developers cannot find the claims-status service?

A. The product that bundles the API was never published to the Developer Hub, so app developers have nothing to discover or subscribe to yet.

B. The API proxy still holds an unresolved error from testing, so the platform hid it from the Developer Hub until the proxy has been corrected.
C. The claims system is refusing the developers' lookups, so the service never appears for them to discover or subscribe to in the Developer Hub.
D. The clinic developers were each missing an account, so the platform blocked them from seeing any service that is listed in the Developer Hub.

Answer: A

Explanation:
An API becomes discoverable to app developers only when the product that bundles it is published to the Developer Hub. The proxy works in testing, but with no product published there is nothing for the clinic developers to find or subscribe to — the offering was simply never made available.

Question: 2

CHALLENGE 1 — Publishing the Claims-Status API So Clinic Developers Can Find It

The developer wants the pilot clinic app developers to be able to discover and subscribe to the working claims-status API.
Which action best makes the claims-status API available to the clinic developers?

A. Rebuild the API proxy in a new package so the platform treats it as a fresh service and lists it for the developers automatically.
B. Publish the product that bundles the claims-status API to the Developer Hub so app developers can discover and subscribe to it.
C. Raise the resources assigned to the proxy so it can handle the clinics and therefore begins appearing for them in the Developer Hub.
D. Send each clinic the technical details of the API by email so their developers can call it without using the Developer Hub at all.

Answer: B

Explanation:
Publishing the product that bundles the API to the Developer Hub is the step that turns a private, working proxy into an offering app developers can find and subscribe to, which is precisely the missing action here.

Question: 3

CHALLENGE 1 — Publishing the Claims-Status API So Clinic Developers Can Find It

After acting to make the service available, the developer wants to confirm the clinic developers can genuinely onboard before inviting the rest.
How should the developer verify that the claims-status API is now properly available?

A. Confirm the proxy still returns a correct status when the developer calls it privately, since a working proxy proves the clinics can now subscribe.
B. Check that the API appears in the developer's own design workspace, which confirms that it is being offered to the clinics through the Developer Hub.
C. Have a pilot clinic developer locate the product in the Developer Hub and complete a subscription, confirming the service is discoverable end to end.
D. Verify that the insurer's service desk is now receiving noticeably fewer calls, since a drop in status calls confirms that the clinics have connected to the new API.

Answer: C

Explanation:
The real test of availability is a clinic developer finding the product in the Developer Hub and subscribing to it; that end-to-end check confirms the offering is discoverable and consumable, which is exactly what was missing before.

Question: 4

CHALLENGE 2 — Selecting the Right Integration Suite Capability for On-Demand Lookups

The team must decide how to deliver an immediate, per-claim status answer to each clinic application. Which SAP Integration Suite capability best fits an on-demand, per-claim status lookup consumed by external clinic applications?

A. A scheduled Cloud Integration flow that periodically copies every claim's status into a single shared file that the clinics can then read on their own.
B. An event stream published through Event Mesh that broadcasts a message to all clinics whenever any claim's status happens to change.
C. A shared spreadsheet the insurer updates through Cloud Integration each morning so the clinics look up a claim from the latest copy.
D. An API exposed through API Management, which offers a governed request-and-response service that returns a claim's status when it is called.

Answer: D

Explanation:
An immediate, per-claim answer is a request-and-response interaction, which is exactly what API Management is designed to expose and govern; a clinic app calls the API for one claim and receives its current status, matching the stated need.

Question: 5

CHALLENGE 2 — Selecting the Right Integration Suite Capability for On-Demand Lookups

A developer proposes a nightly Cloud Integration routine that copies all claim statuses into a shared location, calling it simpler than exposing a live service.

What is the best response to the nightly-batch proposal?

A. A batch copy suits periodic bulk movement, but the need is a live per-claim lookup, so an exposed API fits better and avoids stale, over-shared data.
B. Agree, because a nightly copy is simpler to build and will always show each clinic the fully current status of any claim it happens to look up during that day.
C. Agree, because a nightly batch and an event stream deliver the same result, so either one can stand in for a live claims-status lookup service.
D. Disagree, because only raising the copy routine's throughput could make the shared file update quickly enough to serve immediate claim lookups.

Answer: A

Explanation:
The choice is about matching the capability to the need: a batch copy is for periodic bulk movement, while the requirement is an immediate, single-claim answer, which an exposed API serves directly without leaving clinics reading stale data or holding more member data than they need.
Topic: 2
Micro Skill Drill Exam

Question: 6

Beaubourg Maison runs a chain of home-furnishing stores headquartered in Lyon, France, and uses Cloud Integration to synchronise inventory updates between its stores and a central catalogue. An integration developer is investigating why a few inventory-update messages produced unexpected stock figures and needs to see exactly what each message carried.
In message monitoring the affected messages show a completed status, so they were processed. When the developer opens a completed message to inspect it, however, the message payload is not available for viewing, so the developer cannot see the data that was actually sent. A colleague's flow, built for a different interface, does show payloads for messages in the same time window.
The developer confirms the flow is deployed and processing messages normally and now wants to be able to see the message payload for this flow's messages during monitoring.
Which action lets the developer view the message payload for this flow during monitoring?

A. Add a message-transformation step to the integration flow, because a payload becomes visible in monitoring only once such a step has written that content into the message.
B. Grant the developer a broader monitoring authorization for the tenant, because a completed message keeps its payload hidden until the viewing user holds elevated monitoring access rights.
C. Redeploy the integration flow to the runtime, because a completed message exposes its stored payload for viewing only after the flow has been deployed a further time.
D. Raise the log level configured on the integration flow so message content is captured, because payloads appear in monitoring only when the flow logs at a level that stores them.

Answer: D

Explanation:

Whether a message payload can be viewed in monitoring depends on the log level the flow runs at, because that setting determines how much of the message content is captured and stored. This flow is logging at a level that does not retain payloads, which is why completed messages show no content to inspect. Raising the log level so content is captured makes the payload available at the layer that actually governs it.

Question: 7

Shannonside Pharma manufactures generic medicines at its plant near Cork, Ireland, and is connecting a cloud procurement application to an SAP application using SAP Integration Suite. An integration developer must add a validation step that checks and enriches each purchase order as it flows from the procurement application into the SAP application before the order is posted.
One team member proposes editing the SAP application's core so the validation runs directly inside it, arguing this keeps the check close to where the order is posted. The lead reminds the team that the project is committed to SAP's integration strategy and its clean core approach, which shape where custom logic should live so that the SAP application stays standard and easy to upgrade.
The developer wants to place the validation in a way that aligns with the clean core approach.
Which action places the purchase-order validation in line with the clean core approach?

A. Remove the SAP application's standard posting checks for now, because the clean core approach expects existing core controls to be cleared away before new validation logic is added.
B. Modify the SAP application's core so the validation runs inline, because the clean core approach prefers placing enhancements at the exact point where the order is finally posted.
C. Build the validation as integration logic in SAP Integration Suite, outside the SAP core, because the clean core approach keeps enhancements off the core so it stays upgradable.
D. Copy the SAP application's standard posting code and adjust the copied version, because the clean core approach recommends duplicating core objects before any change is applied to them.

Answer: C

Explanation:
The clean core approach keeps custom enhancements out of the SAP core so the core remains standard and can be upgraded with minimal disruption. Building the purchase-order validation as integration logic in SAP Integration Suite places the check outside the core while still enriching the order in transit. This aligns the enhancement with the integration strategy at the layer clean core intends.

Question: 8

Kitakaze Auto Components produces precision engine parts in Nagoya, Japan, and has built an integration flow in Cloud Integration that pulls the daily production schedule from an on-premise planning system into a cloud quality application. The flow is deployed and runs on schedule, and a step earlier in the same flow that calls another cloud service completes successfully every time.
The step that calls the on-premise planning system, however, fails on every run with a connection error, so the production schedule never reaches the quality application. The cloud-to-cloud step in the same flow keeps working, which tells the developer the flow itself is executing.

The developer confirms that the flow is deployed and that the credentials for the on-premise planning system are correct, and now needs the flow to be able to reach that on-premise system.
Which action lets the integration flow reach the on-premise planning system?

A. Copy the on-premise planning data into the cloud quality application in advance, because a cloud flow can read planning data only when that data no longer resides on-premise.
B. Add a retry with a longer timeout on the failing step, because an on-premise system becomes reachable from a cloud flow once the calling step is set to wait long enough for it.
C. Configure the connectivity option that securely bridges the cloud flow to the on-premise network, because a cloud flow reaches an on-premise system only through that link.
D. Raise the log level on the integration flow for the failing step, because a call to an on-premise system succeeds once that step is logged in enough detail to run to completion.

Answer: C

Explanation:
A cloud integration flow cannot reach a system inside an on-premise network directly; it needs a secure connectivity option configured to bridge the cloud tenant to that network. The cloud-to-cloud step works precisely because it needs no such bridge, while the on-premise step fails for lack of one. Configuring that connectivity link establishes the path at the layer that actually enables the call.

Question: 9

Meridian Freightways coordinates cross-border road haulage from its operations centre in Hamburg, Germany, and uses the API Management capability of SAP Integration Suite to expose shipment-status data to the app teams at its logistics partners. An integration developer has built an API proxy for the shipment-status service, added the required policies, and deployed the proxy to the runtime. Calls made from the developer's own testing tools return shipment data correctly, and the proxy shows as active. The partner app developers, however, report that they cannot find the shipment-status API anywhere in the Developer Hub, so they have no way to subscribe to it or obtain the details needed to call it from their own applications. Internal test traffic works, but nothing is visible to the external app teams.
The developer confirms the proxy is active and the policies are in place, and now wants to make the API discoverable to the partner app developers.
Which action makes the shipment-status API discoverable to the partner app developers?

A. Grant each partner app developer direct runtime permissions on the API proxy, because an API stays hidden from external teams until every partner holds its own runtime access rights.
B. Add the API proxy to a product and publish that product to the Developer Hub, because app developers discover and subscribe to an API only through a published product.
C. Redeploy the API proxy to the runtime a second time, because a proxy becomes visible to external app developers only after it has been deployed more than once.
D. Attach an additional traffic policy to the API proxy, because app developers can locate an API in the Developer Hub only once all of its policies have been fully completed.

Answer: B

Explanation:
App developers discover and subscribe to an API through the Developer Hub, and an API reaches the Developer Hub only when it is bundled into a product and that product is published. The proxy here works internally but was never placed in a published product, so the partner teams have nothing to find. Adding it to a product and publishing that product exposes the API at the layer that actually controls discoverability.

Question: 10

Rift Valley Power distributes electricity to communities around Nairobi, Kenya, and uses the event-driven messaging capability of SAP Integration Suite to publish meter-reading events from its field devices to a billing consumer that turns them into charges. During normal running the events arrive at the billing consumer and are processed without issue.
During a planned maintenance window the billing consumer is taken offline for about an hour. When it is brought back online, the team finds that every meter-reading event published during that hour is missing and was never delivered; only events published after the consumer returned were processed.
The connection between the publisher and the billing consumer was set up using direct messaging.
The developer needs to ensure that meter-reading events published while the billing consumer is offline are still delivered once it comes back online.
Which action ensures events published while the billing consumer is offline are still delivered afterwards?

A. Add a second billing consumer alongside the existing one, because events are preserved during an outage only when more than one consumer is subscribed to receive them.
B. Move the billing consumer into the same region as the event publisher, because events are retained through an outage only when the publisher and the consumer share a single region.
C. Increase how frequently the meter-reading events are published, because events survive a consumer outage only when they are sent often enough for some to overlap the downtime window.
D. Switch the connection from direct messaging to guaranteed messaging, because guaranteed messaging retains published events until the offline consumer returns to receive them.

Answer: D

Explanation:
Direct messaging delivers an event only if a consumer is available at that moment and otherwise drops it, which is why the events sent during the outage were lost. Guaranteed messaging instead retains each published event until the consumer returns and can receive it. Switching to guaranteed messaging changes the delivery guarantee at the layer that actually determines whether offline-window events survive.

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

