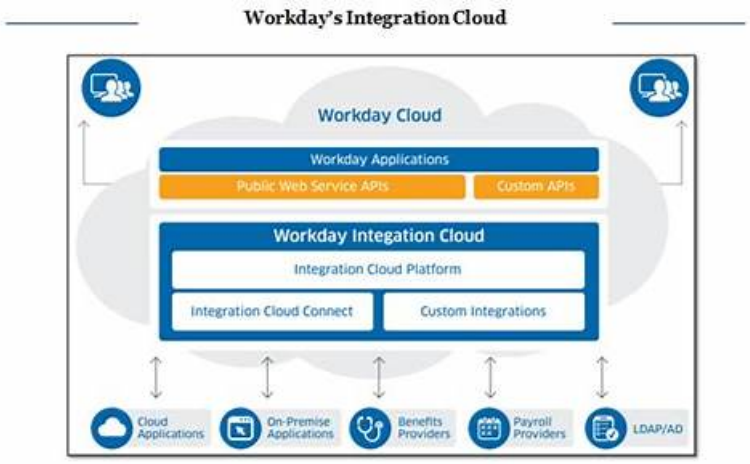
Workday Workday-Pro-Integrationsシュミレーション問題集、Workday-Pro-Integrations模擬試験サンプル



無料でクラウドストレージから最新のCertShiken Workday-Pro-Integrations PDFダンプをダウンロードする：https://drive.google.com/open?id=1QCP2nYDoCE6a3kJUcifPH0-ltCsMPh08

ほとんどの時間インターネットにアクセスできない場合、どこかに行く必要がある場合はオフライン状態ですが、Workday-Pro-Integrations試験のために学習したい場合。心配しないでください、私たちの製品はあなたの問題を解決するのに役立ちます。最新のWorkday-Pro-Integrations試験トレントは、能力を強化し、試験に合格し、認定を取得するのに非常に役立つと確信しています。嫌がらせから抜け出すために、Workday-Pro-Integrations学習教材は高品質で高い合格率を備えています。だから、今すぐ行動しましょう！ Workday-Pro-Integrationsクイズ準備を使用してください。

Workday Workday-Pro-Integrations 認定試験の出題範囲：

トピック	出題範囲
トピック 1	• Reporting: This section of the exam measures the skills of Reporting Analysts and focuses on building, modifying, and managing Workday reports that support integrations. It includes working with report writer tools, custom report types, calculated fields within reports, and optimizing report performance to support automated data exchange.
トピック 2	• Cloud Connect: This section of the exam measures the skills of Workday Implementation Consultants and focuses on using Workday Cloud Connect solutions for third-party integration. It includes understanding pre-built connectors, configuration settings, and how to manage data flow between Workday and external systems while ensuring security and data integrity.
トピック 3	• Calculated Fields: This section of the exam measures the skills of Workday Integration Analysts and covers the creation, configuration, and management of calculated fields used to transform, manipulate, and format data in Workday integrations. It evaluates understanding of field types, dependencies, and logical operations that enable dynamic data customization within integration workflows.
トピック 4	• XSLT: This section of the exam measures the skills of Data Integration Developers and covers the use of Extensible Stylesheet Language Transformations (XSLT) in Workday integrations. It focuses on transforming XML data structures, applying conditional logic, and formatting output for various integration use cases such as APIs and external file delivery.

>> Workday Workday-Pro-Integrationsシュミレーション問題集 <<

Workday-Pro-Integrations模擬試験サンプル、Workday-Pro-Integrations

模擬練習

CertShikenは異なるトレーニングツールと資源を提供してあなたのWorkdayのWorkday-Pro-Integrationsの認証試験の準備にヘルプを差し上げます。編成チュートリアルは授業コース、実践検定、試験エンジンと一部の無料なPDFダウンロードを含めています。

Workday Pro Integrations Certification Exam 認定 Workday-Pro-Integrations 試験問題 (Q51-Q56):

質問 # 51
You are configuring an EIB that uses a custom report as its data source. When attempting to transfer ownership of the report to the Integration System User (ISU), the ISU does not appear as an option for new report owners. You confirm that the ISU already has the necessary access to the report data source and related fields.
Within the Custom Report Creation domain, which security configuration should you update to allow the ISU to appear as a valid report owner?

- A. Assign the ISSG to a row within the Report/Task Permissions table that has View access enabled.
- B. Assign the ISSG to a row within the Integration Permissions table that has Put access enabled.
- C. Assign the ISSG to a row within the Integration Permissions table that has Get access enabled.
- D. Assign the ISSG to a row within the Report/Task Permissions table that has Modify access enabled.

正解：D

解説：
In Workday, for an Integration System User (ISU) to be selectable as a Custom Report Owner, the security group the ISU belongs to must have Modify access to custom reports.
From Workday's security configuration principle:
An ISU does not appear as a valid report owner unless its security group has Modify permission in the Report /Task Permissions section of the Custom Report Creation domain security policy.
This is because report ownership requires write#level access over custom report objects.
Therefore, you must update the Report/Task Permissions table to include the ISSG with Modify access.
Options B, C, and D are incorrect because View or Get/Put do not provide report ownership capabilities.
References:Workday Pro: Integrations - Integration Security and Report Ownership
RulesAdmin#Guide#Authentication#and#Security.pdf - Security Policies & Required Permissions Model

質問 # 52
Refer to the scenario. You are configuring a Core Connector: Worker integration with the Data Initialization Service (DIS) enabled to extract worker demographic and contact information. The integration must include worker fields such as name, address, and a calculated field identifying workers eligible for a phone allowance.
The Phone Allowance Type calculated field exists and is functional in the tenant, but it is not displaying in the output.
What configuration step should you complete to include this field in the output?

- A. Create a Custom Field Override service and reference the calculated field.
- B. Locate the field within the Configure Integration Field Attributes step.
- C. Create a mapping within the Configure Integration Maps step.
- D. Add the calculated field within the Configure Integration Field Overrides step.

正解：B

解説：
In this scenario, a calculated field (Phone Allowance Type) is available and validated in the tenant, but it does not appear in the Core Connector: Worker output. The integration is configured with DIS enabled, and the expected behavior is for all specified worker data - including name, address, and calculated fields - to be included in the output file.
The correct action is to enable the field from the Configure Integration Field Attributes step.
From Workday Pro: Integrations materials:
"In order for a calculated field to be included in a Core Connector output, it must be explicitly located and selected from within the Configure Integration Field Attributes task. This step determines what fields are extracted in the integration output - including any standard or calculated fields available in the object model." Even though the field exists and is functional, it must be manually located within the relevant section (e.g., Worker Data > Compensation or Worker Details), and marked to include in the output.

Incorrect Options Explained:
* A. Configure Integration Field Overrides: This is used to change or override output formatting but does not control field visibility.
* B. Configure Integration Maps: Used for mapping values or converting code sets, not for selecting fields for output.
* C. Create a Custom Field Override service: This is not necessary for simply adding a calculated field; the existing field can be enabled via attributes configuration.
References:
Workday Pro: Core Connector - Field Selection Using Configure Integration Field Attributes Workday Community: How to Include Calculated Fields in Connector Outputs

質問 # 53
What is the task used to upload a new XSLT file for a pre-existing document transformation integration system?

- A. Edit Integration Attachment Service
- B. Edit Integration Service Attachment
- C. Edit Integration Attachment
- D. Edit XSLT Attachment Transformation

正解：D

解説：
In Workday, when you need to upload a new XSLT (Extensible Stylesheet Language Transformations) file to modify or replace an existing transformation within a pre-existing document transformation integration system, the specific task required is "Edit XSLT Attachment Transformation." This task allows users to update the XSLT file that governs how XML data is transformed within the integration system without creating an entirely new transformation object.
Here's why this is the correct answer:
* Workday's integration systems often rely on XSLT to transform XML data into the desired format for downstream systems or processes. When an XSLT file has already been associated with an integration system (e.g., as part of an Enterprise Interface Builder (EIB) or a Document Transformation Connector), updating it requires accessing the existing transformation configuration.
* The "Edit XSLT Attachment Transformation" task enables users to upload a revised version of the XSLT file. This action replaces the previous file while maintaining the integration system's configuration, ensuring continuity without necessitating additional changes to the system itself.
* This task is distinct from other options because it specifically targets the transformation logic (XSLT) rather than broader integration components or services.
Let's examine why the other options are incorrect:
* A. Edit Integration Attachment: This task is used to manage generic attachments associated with an integration, such as input files or supplementary documents, but it does not specifically address XSLT transformations. It lacks the precision required for updating transformation logic.
* B. Edit Integration Attachment Service: This is not a recognized task in Workday's integration framework. It appears to be a conflation of terms and does not align with the documented processes for managing XSLT files.
* D. Edit Integration Service Attachment: While this might suggest modifying an attachment related to an integration service, it is not the correct task for handling XSLT files in a document transformation context. Workday documentation consistently points to "Edit XSLT Attachment Transformation" for this purpose.
The process typically involves:
* Navigating to the integration system in Workday (e.g., via the "Search" bar by entering the integration system name).
* Using the related actions menu to select "Integration System" > "Edit XSLT Attachment Transformation."
* Uploading the new XSLT file, which must comply with Workday's size limitations (e.g., 30 MB for attachments) and be properly formatted.
* Saving the changes, which updates the transformation logic without altering other integration configurations.
This approach ensures that transformations remain aligned with business requirements, such as reformatting data for compatibility with external systems, while leveraging Workday's secure and efficient integration tools.
References:
* Workday Pro Integrations Study Guide: "Configure Integration System - TRANSFORMATION" section, which details the use of XSLT files in document transformations and the associated tasks.
* Workday Documentation: "Enterprise Interface Builder (EIB)" and "Document Transformation Connector" sections, where the "Edit XSLT Attachment Transformation" task is outlined for updating XSLT files.
* Workday Community: Guidance on managing XSLT attachments, confirming this task as the standard method for updating pre-existing transformations.

質問 # 54
What is the purpose of the <xsl:template> element?

- A. Generate an output file name.
- B. Provide rules to apply to a specified node.
- C. Determine the output file type.
- D. Grant access to the XSLT language.

正解：B

解説：
The <xsl:template> element is a fundamental component of XSLT (Extensible Stylesheet Language Transformations), which is widely used in Workday integrations, particularly within document transformation systems such as those configured via the Enterprise Interface Builder (EIB) or Document Transformation Connectors. Its primary purpose is to define rules or instructions that dictate how specific nodes in an XML source document should be processed and transformed into the desired output format.
Here's a detailed explanation of why this is the correct answer:
In XSLT, the <xsl:template> element is used to create reusable transformation rules. It typically includes a match attribute, which specifies the XML node or pattern (e.g., an element, attribute, or root node) to which the template applies. For example, <xsl:template match="Employee"> would target all <Employee> elements in the source XML.
Inside the <xsl:template> element, you define the logic-such as extracting data, restructuring it, or applying conditions-that determines how the matched node is transformed into the output. This makes it a core mechanism for controlling the transformation process in Workday integrations.
In the context of Workday, where XSLT is often used to reformat XML data into formats like CSV, JSON, or custom XML for external systems, <xsl:template> provides the structure for specifying how data from Workday's XML output (e.g., payroll or HR data) is mapped and transformed.
Let's evaluate why the other options are incorrect:
A . Determine the output file type: The <xsl:template> element does not control the output file type (e.g., XML, text, HTML). This is determined by the <xsl:output> element in the XSLT stylesheet, which defines the format of the resulting file independently of individual templates.
B . Grant access to the XSLT language: This option is nonsensical in the context of XSLT. The <xsl:template> element is part of the XSLT language itself and does not "grant access" to it; rather, it is a functional building block used within an XSLT stylesheet.
D . Generate an output file name: The <xsl:template> element has no role in naming the output file. In Workday, the output file name is typically configured within the integration system settings (e.g., via the EIB or connector configuration) and is not influenced by the XSLT transformation logic.
An example of <xsl:template> in action might look like this in a Workday transformation:
<xsl:template match="wd:Worker">
<Employee>
<Name><xsl:value-of select="wd:Worker_Name"/></Name>
</Employee>
</xsl:template>
Here, the template matches the Worker node in Workday's XML schema and transforms it into a simpler <Employee> structure with a Name element, demonstrating its role in providing rules for node transformation.
:
Workday Pro Integrations Study Guide: "Configure Integration System - TRANSFORMATION" section, which explains XSLT usage in Workday and highlights <xsl:template> as the mechanism for defining transformation rules.
Workday Documentation: "XSLT Transformations in Workday" under the Document Transformation Connector, noting <xsl:template> as critical for node-specific processing.
W3C XSLT 1.0 Specification (adopted by Workday): Section 5.3, "Defining Template Rules," which confirms that <xsl:template> provides rules for applying transformations to specified nodes.
Workday Community: Examples of XSLT in integration scenarios, consistently using <xsl:template> for transformation logic.

質問 # 55
Refer to the following XML to answer the question below.
You are an integration developer and need to write XSLT to transform the output of an EIB which is making a request to the Get Job Profiles web service operation. The root template of your XSLT matches on the <wd:
Get_Job_Profiles_Response> element. This root template then applies templates against <wd:Job_Profile>.
What XPath syntax would be used to select the value of the ID element which has a wd:type attribute named Job_Profile_ID when the <xsl:value-of> element is placed within the template which matches on <wd:
Job_Profile>?

- A. wd:Job_Profile_Reference/wd:ID/wd:type='Job_Profile_ID'
- B. wd:Job_Profile_Reference/wd:ID/@wd:type='Job_Profile_ID'
- C. wd:Job_Profile_Reference/wd:ID/[@wd:type='Job_Profile_ID']
- D. wd:Job_Profile_Reference/wd:ID[@wd:type='Job_Profile_ID']

正解：D

解説：
As an integration developer working with Workday, you are tasked with transforming the output of an Enterprise Interface Builder (EIB) that calls the Get_Job_Profiles web service operation. The provided XML shows the response from this operation, and you need to write XSLT to select the value of the <wd:ID> element where the wd:type attribute equals "Job_Profile_ID." The root template of your XSLT matches on
<wd:Get_Job_Profiles_Response> and applies templates to <wd:Job_Profile>. Within this template, you use the <xsl:value-of> element to extract the value. Let's analyze the XML structure, the requirement, and each option to determine the correct XPath syntax.
Understanding the XML and Requirement
The XML snippet provided is a SOAP response from the Get_Job_Profiles web service operation in Workday, using the namespace xmlns:wd="urn:com.workday/bsvc" and version wd:version="v43.0". Key elements relevant to the question include:
* The root element is <wd:Get_Job_Profiles_Response>.
* It contains <wd:Response_Data>, which includes <wd:Job_Profile> elements.
* Within <wd:Job_Profile>, there is <wd:Job_Profile_Reference>, which contains multiple <wd:ID> elements, each with a wd:type attribute:
* <wd:ID wd:type="WID">1740d3eca2f2ed9b6174ca7d2ae88c8c</wd:ID>
* <wd:ID wd:type="Job_Profile_ID">Senior_Benefits_Analyst</wd:ID>
The task is to select the value of the <wd:ID> element where wd:type="Job_Profile_ID" (e.g., "Senior_Benefits_Analyst") using XPath within an XSLT template that matches <wd:Job_Profile>. The <xsl:value-of> element outputs the value of the selected node, so you need the correct XPath path from the <wd:Job_Profile> context to the specific <wd:ID> element with the wd:type attribute value "Job_Profile_ID." Analysis of Options Let's evaluate each option based on the XML structure and XPath syntax rules:
* Option A: wd:Job_Profile_Reference/wd:ID/wd:type='Job_Profile_ID'
* This XPath attempts to navigate from wd:Job_Profile_Reference to wd:ID, then to wd:type='Job_Profile_ID'. However, there are several issues:
* wd:type='Job_Profile_ID' is not valid XPath syntax. In XPath, to filter based on an attribute value, you use the attribute selector [@attribute='value'], not a direct comparison like wd:type='Job_Profile_ID'.
* wd:type is an attribute of <wd:ID>, not a child element or node. This syntax would not select the <wd:ID> element itself but would be interpreted as trying to match a nonexistent child node or property, resulting in an error or no match.
* This option is incorrect because it misuses XPath syntax for attribute filtering.
* Option B: wd:Job_Profile_Reference/wd:ID/@wd:type='Job_Profile_ID'
* This XPath navigates to wd:Job_Profile_Reference/wd:ID and then selects the @wd:type attribute, comparing it to "Job_Profile_ID" with =@wd:type='Job_Profile_ID'. However:
* The =@wd:type='Job_Profile_ID' syntax is invalid in XPath. To filter based on an attribute value, you use [@wd:type='Job_Profile_ID'] as a predicate, not an equality comparison in this form.
* This XPath would select the wd:type attribute itself (e.g., the string "Job_Profile_ID"), not the value of the <wd:ID> element. Since <xsl:value-of> expects a node or element value, selecting an attribute directly would not yield the desired "Senior_Benefits_Analyst" value.
* This option is incorrect due to the invalid syntax and inappropriate selection of the attribute instead of the element value.
* Option C: wd:Job_Profile_Reference/wd:ID[@wd:type='Job_Profile_ID']
* This XPath navigates from wd:Job_Profile_Reference to wd:ID and uses the predicate [@wd:type='Job_Profile_ID'] to filter for <wd:ID> elements where the wd:type attribute equals "Job_Profile_ID."
* In the XML, <wd:Job_Profile_Reference> contains:
* <wd:ID wd:type="WID">1740d3eca2f2ed9b6174ca7d2ae88c8c</wd:ID>
* <wd:ID wd:type="Job_Profile_ID">Senior_Benefits_Analyst</wd:ID>
* The predicate [@wd:type='Job_Profile_ID'] selects the second <wd:ID> element, whose value is "Senior_Benefits_Analyst."
* Since the template matches <wd:Job_Profile>, and <wd:Job_Profile_Reference> is a direct child of <wd:Job_Profile>, this path is correct:
* <wd:Job_Profile> # <wd:Job_Profile_Reference> # <wd:ID[@wd:type='Job_Profile_ID']>.
* When used with <xsl:value-of select="wd:Job_Profile_Reference/wd:ID[@wd:type='Job_Profile_ID']"/>, it outputs "Senior_Benefits_Analyst," fulfilling the requirement.

* This option is correct because it uses proper XPath syntax for attribute-based filtering and selects the desired <wd:ID> value.
* Option D: wd:Job_Profile_Reference/wd:ID/[@wd:type='Job_Profile_ID']
* This XPath is similar to Option C but includes an extra forward slash before the predicate: wd:ID/[@wd:type='Job_Profile_ID']. In XPath, predicates like [@attribute='value'] are used directly after the node name (e.g., wd:ID[@wd:type='Job_Profile_ID']), not separated by a slash. The extra slash is syntactically incorrect and would result in an error or no match, as it implies navigating to a child node that doesn't exist.
* This option is incorrect due to the invalid syntax.
Why Option C is Correct
Option C, wd:Job_Profile_Reference/wd:ID[@wd:type='Job_Profile_ID'], is the correct XPath syntax because:
* It starts from the context node <wd:Job_Profile> (as the template matches this element) and navigates to <wd:Job_Profile_Reference/wd:ID>, using the predicate [@wd:type='Job_Profile_ID'] to filter for the <wd:ID> element with wd:type="Job_Profile_ID".
* It correctly selects the value "Senior_Benefits_Analyst," which is the content of the <wd:ID> element where wd:type="Job_Profile_ID".
* It uses standard XPath syntax for attribute-based filtering, aligning with Workday's XSLT implementation for web service responses.
* When used with <xsl:value-of>, it outputs the required value, fulfilling the question's requirement.
Practical Example in XSLT
Here's how this might look in your XSLT:
<xsl:template match="wd:Job_Profile">
<xsl:value-of select="wd:Job_Profile_Reference/wd:ID[@wd:type='Job_Profile_ID']"/>
</xsl:template>
This would output "Senior_Benefits_Analyst" for the <wd:ID> element with wd:type="Job_Profile_ID" in the XML.
Verification with Workday Documentation
The Workday Pro Integrations Study Guide and SOAP API Reference (available via Workday Community) detail the structure of the Get_Job_Profiles response and how to use XPath in XSLT for transformations. The XML structure shows <wd:Job_Profile_Reference> containing <wd:ID> elements with wd:type attributes, and the guide emphasizes using predicates like [@wd:type='value'] to filter based on attributes. This is a standard practice for navigating Workday web service responses.
Workday Pro Integrations Study Guide References
* Section: XSLT Transformations in EIBs - Describes using XSLT to transform web service responses, including selecting elements with XPath and attribute predicates.
* Section: Workday Web Services - Details the Get_Job_Profiles operation and its XML output structure, including <wd:Job_Profile_Reference> and <wd:ID> with wd:type attributes.
* Section: XPath Syntax - Explains how to use predicates like [@wd:type='Job_Profile_ID'] for attribute- based filtering in Workday XSLT.
* Workday Community SOAP API Reference - Provides examples of XPath navigation for Workday web service responses, including attribute selection.
Option C is the verified answer, as it correctly selects the <wd:ID> value with wd:type="Job_Profile_ID" using the appropriate XPath syntax within the <wd:Job_Profile> template context.

質問 # 56
.....

- Workday-Pro-Integrations学習体験談 □ Workday-Pro-Integrations資格問題対応 □ Workday-Pro-Integrations復習資料 □ 今すぐ[www.goshiken.com]で□ Workday-Pro-Integrations □を検索して、無料でダウンロードしてくださいWorkday-Pro-Integrations無料問題
- Workday-Pro-Integrations無料問題 □ Workday-Pro-Integrations無料模擬試験 □ Workday-Pro-Integrations出題範囲 □ ウェブサイト➠ www.passtest.jp □から【 Workday-Pro-Integrations 】を開いて検索し、無料でダウンロードしてくださいWorkday-Pro-Integrations日本語版サンプル
- 有効的なWorkday-Pro-Integrationsシュミレーション問題集 - 合格スムーズWorkday-Pro-Integrations模擬試験サンプル | 最高のWorkday-Pro-Integrations模擬練習 □ Open Webサイト[www.goshiken.com]検索「 Workday-Pro-Integrations 」無料ダウンロードWorkday-Pro-Integrations対応問題集
- 有効的なWorkday-Pro-Integrationsシュミレーション問題集 - 合格スムーズWorkday-Pro-Integrations模擬試験サンプル | 最高のWorkday-Pro-Integrations模擬練習 □☀ www.passtest.jp □☀□サイトにて最新➠ Workday-Pro-Integrations □問題集をダウンロードWorkday-Pro-Integrations学習体験談
- ハイパスレートWorkday Workday-Pro-Integrations | 最高のWorkday-Pro-Integrationsシュミレーション問題集試験 | 試験の準備方法Workday Pro Integrations Certification Exam模擬試験サンプル □ URL ✔ www.goshiken.com □✔□をコピーして開き、（ Workday-Pro-Integrations ）を検索して無料でダウンロードしてくださいWorkday-Pro-Integrationsテストサンプル問題
- 試験の準備方法-素敵なWorkday-Pro-Integrationsシュミレーション問題集試験-認定するWorkday-Pro-Integrations模擬試験サンプル □▸ www.xhs1991.com◂に移動し、{ Workday-Pro-Integrations }を検索して無料でダウンロードしてくださいWorkday-Pro-Integrationsウェブトレーニング
- Workday-Pro-Integrations無料問題 □ Workday-Pro-Integrations関連日本語版問題集 □ Workday-Pro-Integrations無料模擬試験 □“ www.goshiken.com ”を開き、▸ Workday-Pro-Integrations ◂を入力して、無料でダウンロードしてくださいWorkday-Pro-Integrations試験復習赤本
- 素晴らしいWorkday-Pro-Integrations | 権威のあるWorkday-Pro-Integrationsシュミレーション問題集試験 | 試験の準備方法Workday Pro Integrations Certification Exam模擬試験サンプル □ ➤ www.japancert.com □サイトで【 Workday-Pro-Integrations 】の最新問題が使えるWorkday-Pro-Integrations学習体験談
- www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, bbs.t-firefly.com, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, Disposable vapes

無料でクラウドストレージから最新のCertShiken Workday-Pro-Integrations PDFダンプをダウンロードする：https://drive.google.com/open?id=1QCP2nYDoCE6a3kJUcifPH0-ltCsMPh08