

## Unit 1: Study Up on Debugging and Error Handling

### Learning Objectives

After completing this unit, you'll be able to:

- Handle errors properly given a scenario.
- Use the console and breakpoints given code to be debugged.

### Key Topics

This unit prepares you for the Debugging and Error Handling section of the Salesforce JavaScript Developer I multiple-choice exam, which makes up 7% of the overall exam. This section of the exam tests these topics:

- Exceptions
- Selective catching
- Assertions
- Breakpoints
- 

This unit provides a number of interactive, real-world, scenario-based questions that are a lot like the ones you'll encounter as a Salesforce JavaScript Developer. Looking at these scenarios helps prepare you to take this Debugging and Error Handling section of the Salesforce JavaScript Developer I multiple-choice exam. As you tackle the practice questions, you get immediate feedback on your answers, along with detailed information on why your answers are correct (or incorrect).

The unit also contains interactive flashcards centered around study topics that help you prepare for the Debugging and Error Handling section of the exam.

### Download the Guide

Would you like a hard copy of this module's content as a study aid? Download the [JavaScript Developer I Certification Prep: Debugging, Error Handling, and Testing guide](#). (Each module in this trail includes a link to a printable version of the content that you can download.)

### Exam Practice Questions

Ready to jump in? The sample tool below is not scored—it's just an easy way to quiz yourself. To use it, read the scenario, then click on the answer you think is correct. Some questions may have more than one correct answer. Click **Submit** to learn whether the answer you chose is correct or incorrect, and why; if there's a longer explanation, click  and then click anywhere in the window to close it. When you reach the end, you can review the answers or retake the questions.

# JavaScript Dev I Cert Prep: Debugging, Error Handling, Testing

## Question 1

What are the two main properties of an error object that's passed as an argument to catch in a try...catch construct?

Option		Feedback
A	name and stacktrace	Incorrect. While name is correct, stacktrace isn't.
B	title and message	Incorrect. The error object does not have a title property.
C	title and stack	Incorrect. The error object does not have a title property and stack is a nonstandard property.
D	name and message	<b>Correct. Both name and message are main properties.</b>

# JavaScript Dev I Cert Prep: Debugging, Error Handling, Testing

## Question 2

A developer wants to set a breakpoint in his code while in the editor so they don't have to switch to the browser. What is the in-line command for setting a breakpoint?

Option		Feedback
A	break	Incorrect. The break statement is used in switch statements or loops.
B	debugger	<b>Correct. The debugger command will cause a breakpoint.</b>
C	debug	Incorrect. Debug is not a command.
D	breakpoint	Incorrect. Breakpoint is not a command.

Did you get a scenario wrong? Check out the table below for related study material.

Scenario 1	Review <a href="#">JavaScript.info: Error handling, try...catch</a> documentation to understand the try...catch construct and its usage.
Scenario 2	Study up on using breakpoints using the <a href="#">Javascript.info: Debugging in Chrome documentation</a> .

# JavaScript Dev I Cert Prep: Debugging, Error Handling, Testing

## Exam Topic Flashcards

The following flashcards cover debugging and error handling. Use these interactive flashcards to brush up on some of the key topics you'll find on this part of the exam.

Read the question or term on each card, then click or tap the card to reveal the correct answer. Click the right-facing arrow to move to the next card, and the left-facing arrow to return to the previous card.

Card 1	
Front of card (term or question)	What are some benefits of using breakpoints?
Back of card (definition or answer)	JavaScript will stop executing at each breakpoint to let you examine the current JavaScript values. After examining the values, you can resume the execution of code. You can even step from breakpoint to breakpoint.

Card 2	
Front of card (term or question)	What code block can be executed regardless of whether an exception was thrown or caught in a try...catch block.
Back of card (definition or answer)	The finally block can be used to execute code at the end of a try...catch block. It will run even if no catch block handles the exception.

Did you get a flashcard wrong? Check out the table below for related study material.

Flashcard 1	Study up on using breakpoints by reviewing <a href="#">JavaScript.info: Debugging</a> .
Flashcard 2	Learn how to describe the options for handling errors with the try...catch block by reviewing <a href="#">MDN web docs: try...catch</a> .

# JavaScript Dev I Cert Prep: Debugging, Error Handling, Testing

## Related Badges

Looking for more information? Explore these related badges.

Badge	Content Type
	Module
	Module

You've reviewed debugging and error handling. Next, let's take a look at testing.

## Resources

- [External Site: JavaScript.info: Debugging in Chrome](#)
- [External Site: JavaScript.info: Error handling](#)

## Unit 2: Explore Testing

### Learning Objectives

After completing this unit, you'll be able to:

- Understand differences between unit testing, integration testing, and end-to-end testing.
- Describe some of the tools used to perform JavaScript testing.

### Key Topics

This unit prepares you for the Testing section of the Salesforce JavaScript Developer I multiple-choice exam, which makes up 7% of the overall exam. This section of the exam tests these topics.

- Test types
- Running tests
- Testing tools
- 

Like the previous unit, this unit contains practice scenario-based questions and flashcards.

### Exam Practice Questions

Ready to jump in? The sample tool below is not scored—it's just an easy way to quiz yourself. To use it, read the scenario, then click on the answer you think is correct. Some questions may have more than one correct answer. Click **Submit** to learn whether the answer you chose is correct or incorrect, and why; if there's a longer explanation, click  and then click anywhere in the window to close it. When you reach the end, you can review the answers or retake the questions.

# JavaScript Dev I Cert Prep: Debugging, Error Handling, Testing

## Question 1

Given the code below:

```
01 let res = sumArr([2,3,4]);
```

which of the following code lines will assert the sumArr method adds the numbers in the array passed in?

Option		Feedback
A	console.assert(res === 9);	<b>Correct. The console.assert method writes an error message to the console if the assertion is false. If the assertion is true, nothing happens.</b>
B	console.log(res === 9);	Incorrect. The console.log method will write whatever is passed in to the console.
C	console.assert(res != 9);	Incorrect. This console.assert will return false. It should be written to return true so it will only display in the console when it is false.
D	console.error(res != 9);	Incorrect. The console.error method will write whatever is passed in to the console as an error.

# JavaScript Dev I Cert Prep: Debugging, Error Handling, Testing

## Question 2

Given the code below:

```
01 let res = sum3([1, 2, 3]);  
02 console.assert(res === 6);
```

The sum3 method gets updated to multiply the numbers instead of adding them. Line 02 is now a false-positive assertion. How can the test be changed to fix it?

Option		Feedback
A	<pre>01 let res = sum3([1, 2, 3, 4]); 02 console.assert(res === 10);</pre>	Incorrect. Since the method now multiplies the numbers, 10 is not correct.
B	<pre>01 let res = sum3([1, 2]); 02 console.assert(res === 3);</pre>	Incorrect. Even by reducing the numbers, the result is still a multiplication of the numbers resulting in 2 as the result.
C	<pre>01 let res = sum3([1, 2, 3, 4]); 02 console.assert(res === 24);</pre>	<b>Correct. Adding another number in the sequence ensures the result is different if added or multiplied.</b>
D	<pre>01 let res = sum3([3, 2, 1]); 02 console.assert(res === 6);</pre>	Incorrect. Just changing the arrangement of the numbers doesn't change the result being the same for both adding them and multiplying them.

Did you get a scenario wrong? Check out the table below for related study material.

Scenario 1	Review the <a href="#">MDN web docs: console.assert() documentation</a> to better understand the console.assert method and its properties.
Scenario 2	Discover testing issues by reviewing <a href="#">Kent C. Dodds: Testing Implementation Details</a> .

# JavaScript Dev I Cert Prep: Debugging, Error Handling, Testing

## Exam Topic Flashcards

The following flashcards cover Testing. Use these interactive flashcards to brush up on some of the key topics you'll find on this part of the exam.

Read the question or term on each card, then click or tap the card to reveal the correct answer. Click the right-facing arrow to move to the next card, and the left-facing arrow to return to the previous card.

Card 1	
Front of card (term or question)	If black-box testing treats software under test as a black-box without knowing its internals. What is white-box testing?
Back of card (definition or answer)	White-box testing is fully aware of the internal code that is being tested and uses that as part of the testing.

Card 2	
Front of card (term or question)	A developer wrote a method and created some test assertions for it. Another developer has now updated the method and the tests are still passing. Is this enough or should the tests be updated.
Back of card (definition or answer)	The test should be verified for possible false positives. New tests should be added to assert the way the method was updated.

Did you get a flashcard wrong? Check out the table below for related study material.

Flashcard 1	Study up on black-box and white-box testing practices by reviewing <a href="#">Technology Conversations: Black-box vs White-box Testing</a> .
Flashcard 2	Understand false-positive and false-negative testing by reading <a href="#">Kent C. Dodds: Testing Implementation Details</a> .

# JavaScript Dev I Cert Prep: Debugging, Error Handling, Testing

## Related Badges

Looking for more information? Explore these related badges.

Badge	Content Type
 A circular badge icon with a purple gradient background. Inside, there's a stylized purple cube with three faces visible, each showing a different number: <1> on the top face, <2> on the bottom-left face, and <3> on the bottom-right face.	Module
 A circular badge icon with a yellow gradient background. Inside, there's a stylized graphic of interconnected nodes (dots) and arrows, representing a network or data flow.	Module

Congratulations. You've covered 14% of the Salesforce JavaScript Developer I multiple-choice exam material in this badge.

You've reviewed these sections.

- Debugging and Error Handling
- Testing

Be sure to review the other two Salesforce JavaScript Developer I Certification Prep badges. Good luck on your exam!

## Resources

- [External Site: Jest.io: Getting started](#)

## Unit 3: Review Server Side JavaScript

### Learning Objectives

After completing this unit, you'll be able to:

- Given a scenario and requirements, infer which Node.js implementation is a good solution.
- Given a scenario and requirements, infer which Node.js CLI command is a good solution.
- Know the core Node.js modules and given requirements, infer which Node.js library/framework is a good solution.
- Given a scenario and requirements, distinguish which Node.js Package Management solution is the most fitting.

### Key Topics

This unit prepares you for the Server-Side JavaScript section of the Salesforce JavaScript Developer I multiple-choice exam, which makes up 8% of the overall exam. This section of the exam tests these topics:

- Node.js fundamentals
- Package files
- File system and HTTP modules
- Streams

Like the previous units, this unit contains practice scenario-based questions and flashcards.

### Exam Practice Questions

Ready to jump in? The sample tool below is not scored—it's just an easy way to quiz yourself. To use it, read the scenario, then click on the answer you think is correct. Some questions may have more than one correct answer. Click **Submit** to learn whether the answer you chose is correct or incorrect, and why; if there's a longer explanation, click  and then click anywhere in the window to close it. When you reach the end, you can review the answers or retake the questions.

# JavaScript Dev I Cert Prep: Debugging, Error Handling, Testing

## Question 1

Here is the package.json for the bar.awesome module:

```
{"name": "bar.awesome", "version": "1.3.5", "peerDependencies": { "baz": "5.x" }}
```

A particular project has the package.json definition below.

```
{"name": "UC Project Extra", "version": "0.0.5", "dependencies": { "bar.awesome": "1.3.5", "baz": "6.0.0" }}
```

What happens when a developer executes npm install?

Option		Feedback
A	The command fails because bar.awesome does not have any dependency.	Incorrect. bar does have a dependency on baz.
B	The command fails because bar versions are not compatible.	Incorrect. The bar versions are compatible, but the baz versions are not.
C	The command succeeds but displays a warning about a version mismatch.	<b>Correct. There is a mismatch on the baz dependency that causes the warning.</b>
D	The command succeeds with no errors or warnings.	Incorrect. The command does succeed but there is a warning for the mismatched baz dependency.

# JavaScript Dev I Cert Prep: Debugging, Error Handling, Testing

## Question 2

Refer to the code below:

```
01 const https = require('https');
02 const server = https.createServer((req, res) => {
03   // code goes here
04   let reqData = JSON.parse(chunk);
05   console.log(reqData);
06 });
07 res.end('OK');
08 });
09 server.listen(8000);
```

Which code does the developer need to add to line 03 to receive incoming request data?

Option		Feedback
A	<code>req.on('data', (chunk) =&gt;</code> <code>{</code>	<b>Correct. The chunk argument is passed in for JSON.parse to use.</b>
B	<code>req.get((chunk) =&gt; {</code>	Incorrect. The get method is used with http requests.
C	<code>req.data((chunk) =&gt; {</code>	Incorrect. Data is what is getting passed in, not a method.
D	<code>req.on('get', (chunk) =&gt; {</code>	Incorrect. The get argument is not passed in so can not be used this way.

Did you get a scenario wrong? Check out the table below for related study material.

Scenario 1	Review the package.json configuration and dependencies issues by reading the <a href="#">npm package.json documentation</a> .
Scenario 2	Describe the server call functions by reviewing the <a href="#">Node.js https documentation</a> .

# JavaScript Dev I Cert Prep: Debugging, Error Handling, Testing

## Exam Topic Flashcards

The following flashcards cover server-side JavaScript. Use these interactive flashcards to brush up on some of the key topics you'll find on this part of the exam.

Read the question or term on each card, then click or tap the card to reveal the correct answer. Click the right-facing arrow to move to the next card, and the left-facing arrow to return to the previous card.

Card 1	
Front of card (term or question)	What are the most important items in your npm package.json file if you plan to publish a package?
Back of card (definition or answer)	The name and version. Together they form a unique identifier for the package.

Card 2	
Front of card (term or question)	When it comes to frameworks, what is one of the main benefits of using popular frameworks?
Back of card (definition or answer)	One of the main benefits is a strong community supporting the framework. This usually results in good documentation and resources.

Did you get a flashcard wrong? Check out the table below for related study material.

Flashcard 1	Study up on the npm package.json settings by reviewing the <a href="#">Specifics of npm's package.json handling</a> .
Flashcard 2	Discover the popular frameworks that are available by reviewing <a href="#">the best front-end frameworks</a> and <a href="#">the best back-end frameworks</a> .

# JavaScript Dev I Cert Prep: Debugging, Error Handling, Testing

## Related Badges

Looking for more information? Explore these related badges.

Badge	Content Type
 A circular badge icon featuring a purple cube with three arrows pointing outwards from its faces, labeled <1>, <2>, and <3>.	Module
 A circular badge icon featuring a yellow background with a green border, containing a network graph of nodes connected by lines.	Module
 A circular badge icon featuring a purple background with a blue border, containing a 3D cube composed of smaller cubes.	Module

Congratulations. You've covered 38% of the JavaScript Developer I multiple-choice exam material in this badge.

You've reviewed these sections.

- Browsers and Events
- Asynchronous Programming
- Server-Side JavaScript

Be sure to review the other two JavaScript Developer I Certification Prep badges. Good luck on your exam!

## Resources

- [External Site: Node: Documentation](#)
- [External Site: npm: CLI documentation](#)