



Amazon Q Developer SDLC Workshop

Michael Lin

Sr. Solutions Architect
Amazon Web Services

[Workshop Studio](#) > [Sign in](#)

Sign in

Choose a preferred sign-in method

Email one-time password (OTP)

Enter your personal or corporate email to receive a one-time password

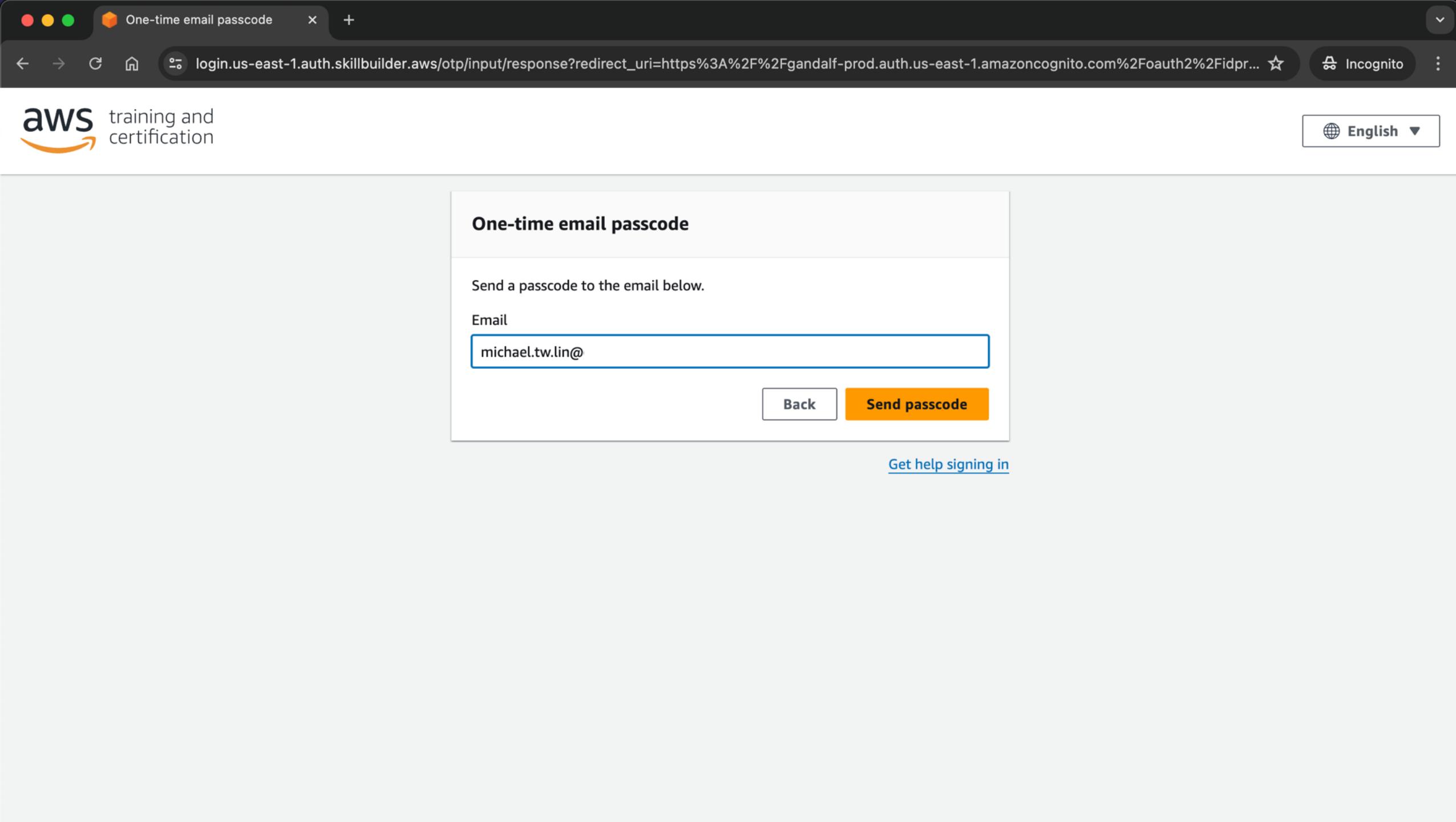
AWS Builder ID

Login with AWS Builder ID, a new personal profile for builders

Amazon employee

Login with your Amazon Corporate account. Only for Amazon Employees.





Verify one-time email passcode

← → ⌛ ⌂ login.us-east-1.auth.skillbuilder.aws/otp/challenge?redirect_uri=https%3A%2F%2Fgandalf-prod.auth.us-east-1.amazoncognito.com%2Foauth2%2Fidprespo... ☆ Incognito :

aws training and certification

English ▾

One-time email passcode

We sent a passcode to michael.tw.lin@gmail.com. You should receive it within 5 minutes.

Passcode (9-digit) [Resend passcode](#)

529717102

Back

Sign in

[Get help signing in](#)

- Step 1
Enter event access code
- Step 2
Review and join

Review and join

Event details

Name
TCC-Bedrock-Dryrun

Start time
7/27/2024 03:02 PM

Duration
72 hours

Level
300

Description
TCC-Bedrock-Dryrun

Terms and Conditions

Read and accept before joining the event

Read and accept before joining the event:

1. By using AWS Workshop Studio for the relevant event, You agree to [the AWS Event Terms and Conditions](#), the [AWS Responsible AI Policy](#), and the [AWS Acceptable Use Policy](#).
2. If You are under 18 years old, you may participate in the relevant event using AWS Workshop Studio: (a) if You are at least the minimum age below based on the country or region in which You reside, and (b) with the involvement of a parent, guardian, or educator.



Country or region	Minimum age
All countries or regions not listed below (including the United States, Brazil, the United Kingdom, and India)	13
Canada, China, Republic of Korea (South Korea)	14
Australia	15

catalog.us-east-1.prod.workshops.aws/join

aws workshop studio

Philippines, Thailand, Turkey, and countries in Africa

3. You acknowledge and agree that You are using an AWS-owned account that You will only be able to access during the relevant event. You have no ownership rights over this AWS-owned account.

4. During the relevant event, while using this AWS-owned account, You will not use, import, input, or introduce any data, dataset, or other material that contains personal data, financial information, or any other data or materials that may be subject to laws and regulations (such as the General Data Protection Regulation or The Health Insurance Portability and Accountability Act of 1996).

5. If You find residual resources or materials in this AWS-owned account, You will notify your Event Operator immediately.

6. AWS, its affiliates, and any entities or persons acting on AWS's behalf reserves the right to terminate this AWS-owned account and to delete its contents at any time, without any notice to You.

7. During the relevant event, while using this AWS-owned account, You will not process or run any operation on any data other than test datasets or lab materials that have been approved by AWS.

8. You will not copy, import, export or otherwise create derivative works of materials provided by AWS for use outside of the relevant event.

9. AWS, its affiliates, and any entities or persons acting on AWS's behalf have no obligation to enable the transmission of Your materials through AWS Workshop Studio, and may, in their discretion, edit, block, refuse to post, or remove Your materials at any time, without notice to You.

10. If You access and use a service and/or third-party models that have their own terms during the relevant event, while in the AWS-owned account, You agree to review those terms and comply with them during the event.

11. If You are an AWS Partner using AWS Workshop Studio as part of Your participation in the AWS Partner Network Program, Your use of AWS Workshop Studio is governed by these terms, the AWS Partner Network Terms and Conditions, and the AWS Customer Agreement or other agreement with us governing your use of AWS Services.

12. Your use of AWS Workshop Studio will comply with these terms and all applicable laws. If You fail to comply with any of these terms, Your access to AWS Workshop Studio may be immediately terminated, without notice to You.

I agree with the Terms and Conditions

Cancel Previous Join event

catalog.us-east-1.prod.workshops.aws/join

aws workshop studio

Philippines, Thailand, Turkey, and countries in Africa

3. You acknowledge and agree that You are using an AWS-owned account that You will only be able to access during the relevant event. You have no ownership rights over this AWS-owned account.

4. During the relevant event, while using this AWS-owned account, You will not use, import, input, or introduce any data, dataset, or other material that contains personal data, financial information, or any other data or materials that may be subject to laws and regulations (such as the General Data Protection Regulation or The Health Insurance Portability and Accountability Act of 1996).

5. If You find residual resources or materials in this AWS-owned account, You will notify your Event Operator immediately.

6. AWS, its affiliates, and any entities or persons acting on AWS's behalf reserves the right to terminate this AWS-owned account and to delete its contents at any time, without any notice to You.

7. During the relevant event, while using this AWS-owned account, You will not process or run any operation on any data other than test datasets or lab materials that have been approved by AWS.

8. You will not copy, import, export or otherwise create derivative works of materials provided by AWS for use outside of the relevant event.

9. AWS, its affiliates, and any entities or persons acting on AWS's behalf have no obligation to enable the transmission of Your materials through AWS Workshop Studio, and may, in their discretion, edit, block, refuse to post, or remove Your materials at any time, without notice to You.

10. If You access and use a service and/or third-party models that have their own terms during the relevant event, while in the AWS-owned account, You agree to review those terms and comply with them during the event.

11. If You are an AWS Partner using AWS Workshop Studio as part of Your participation in the AWS Partner Network Program, Your use of AWS Workshop Studio is governed by these terms, the AWS Partner Network Terms and Conditions, and the AWS Customer Agreement or other agreement with us governing your use of AWS Services.

12. Your use of AWS Workshop Studio will comply with these terms and all applicable laws. If You fail to comply with any of these terms, Your access to AWS Workshop Studio may be immediately terminated, without notice to You.

 I agree with the Terms and Conditions

Cancel Previous Join event

Philippines, Thailand, Turkey, and countries in Africa

3. You acknowledge and agree that You are using an AWS-owned account that You will only be able to access during the relevant event. You have no ownership rights over this AWS-owned account.
 4. During the relevant event, while using this AWS-owned account, You will not use, import, input, or introduce any data, dataset, or other material that contains personal data, financial information, or any other data or materials that may be subject to laws and regulations (such as the General Data Protection Regulation or The Health Insurance Portability and Accountability Act of 1996).
 5. If You find residual resources or materials in this AWS-owned account, You will notify your Event Operator immediately.
 6. AWS, its affiliates, and any entities or persons acting on AWS's behalf reserves the right to terminate this AWS-owned account and to delete its contents at any time, without any notice to You.
 7. During the relevant event, while using this AWS-owned account, You will not process or run any operation on any data other than test datasets or lab materials that have been approved by AWS.
 8. You will not copy, import, export or otherwise create derivative works of materials provided by AWS for use outside of the relevant event.
 9. AWS, its affiliates, and any entities or persons acting on AWS's behalf have no obligation to enable the transmission of Your materials through AWS Workshop Studio, and may, in their discretion, edit, block, refuse to post, or remove Your materials at any time, without notice to You.
 10. If You access and use a service and/or third-party models that have their own terms during the relevant event, while in the AWS-owned account, You agree to review those terms and comply with them during the event.
 11. If You are an AWS Partner using AWS Workshop Studio as part of Your participation in the AWS Partner Network Program, Your use of AWS Workshop Studio is governed by these terms, the AWS Partner Network Terms and Conditions, and the AWS Customer Agreement or other agreement with us governing your use of AWS Services.
 12. Your use of AWS Workshop Studio will comply with these terms and all applicable laws. If You fail to comply with any of these terms, Your access to AWS Workshop Studio may be immediately terminated, without notice to You.

I agree with the Terms and Conditions

[Cancel](#)

Previous

[Join event](#)



Q Developer IDE Plugin



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.



qwords-20250527

Amazon Q Developer Workshop -
Building the Q-Words App

Introduction

▼ Setup and Configuration

▶ Running the Workshop at an
AWS Guided Event

Configure your IDE

▶ Running the Workshop in your
own AWS Account

Amazon Q Authentication Setup

(Optional) Additional Languages

▶ Code Comprehension and
Summarization▶ Code Debugging, Testing, and
Optimization

▶ Natural Language to New Feature

▶ Application Modernization and

▼ AWS account access

[Open AWS console
\(us-east-1\)](#)[Get AWS CLI credentials](#)

▶ Content preferences

Exit event

Event information

Start time

5/27/2025 09:20 PM

Duration

72 hours

Accessible regions

us-east-1

Description

qwords-20250527

Workshop

[Get started >](#)

Title

Amazon Q Developer Workshop -
Q-Words

Complexity level

300

AWS services

Amazon CodeWhisperer, Amazon Q

Topics

Builder Experience, Generative AI

Description

This workshop is a hands-on introduction to Amazon Q. Participants will learn how to use Amazon Q to perform a variety of tasks. The workshop is designed for developers of all skill levels, and no prior experience with Amazon Q is required.

Event Outputs (2)



< 1 >



Search

Key



Value



Stack name



Description



Type



Password

sTBRqi65d4vyBLje

code-server

Code-server Password

output

URL

<https://d2v4mt5uwdherk.cloudfront.net/?folder=/Workshop/java>

code-server

Code-server URL

output





qwords-20250527

Amazon Q Developer Workshop -
Building the Q-Words App

Introduction

▼ Setup and Configuration

▶ Running the Workshop at an
AWS Guided Event

Configure your IDE

▶ Running the Workshop in your
own AWS Account

Amazon Q Authentication Setup

(Optional) Additional Languages

▶ Code Comprehension and
Summarization▶ Code Debugging, Testing, and
Optimization

▶ Natural Language to New Feature

▶ Application Modernization and

▼ AWS account access

[Open AWS console
\(us-east-1\)](#)[Get AWS CLI credentials](#)

▶ Content preferences

Exit event

Event information

Start time

5/27/2025 09:20 PM

Duration

72 hours

Accessible regions

us-east-1

Description

qwords-20250527

Workshop

[Get started >](#)

Title

Amazon Q Developer Workshop -
Q-Words

Complexity level

300

AWS services

Amazon CodeWhisperer, Amazon Q

Topics

Builder Experience, Generative AI

Description

This workshop is a hands-on introduction to Amazon Q. Participants will learn how to use Amazon Q to perform a variety of tasks. The workshop is designed for developers of all skill levels, and no prior experience with Amazon Q is required.

Event Outputs (2)



< 1 >



Search

Key

Copied!



Stack name



Description



Type



Type

Password

 sTBRqi65d4vyBLje

code-server

Code-server Password

output

URL

 <https://d2v4mt5uwdherk.cloudfront.net/?folder=/Workshop/java>

code-server

Code-server URL

output





qwords-20250527

Amazon Q Developer Workshop -
Building the Q-Words App

Introduction

▼ Setup and Configuration

▶ Running the Workshop at an
AWS Guided Event

Configure your IDE

▶ Running the Workshop in your
own AWS Account

Amazon Q Authentication Setup

(Optional) Additional Languages

▶ Code Comprehension and
Summarization▶ Code Debugging, Testing, and
Optimization

▶ Natural Language to New Feature

▶ Application Modernization and

▼ AWS account access

[Open AWS console
\(us-east-1\)](#)[Get AWS CLI credentials](#)

▶ Content preferences

Exit event

Event information

Start time

5/27/2025 09:20 PM

Duration

72 hours

Accessible regions

us-east-1

Description

qwords-20250527

Workshop

[Get started >](#)

Title

Amazon Q Developer Workshop -
Q-Words

Complexity level

300

AWS services

Amazon CodeWhisperer, Amazon Q

Topics

Builder Experience, Generative AI

Description

This workshop is a hands-on introduction to Amazon Q. Participants will learn how to use Amazon Q to perform a variety of tasks. The workshop is designed for developers of all skill levels, and no prior experience with Amazon Q is required.

Event Outputs (2)



< 1 >



Search

Key

Copied!



Stack name

Description



Type

Type

Password

[Copied!](#)

code-server

Code-server Password

output

URL

[Copied!](https://d2v4mt5uwdherk.cloudfront.net/?folder=/Workshop/java)

code-server

Code-server URL

output



Welcome to code-server

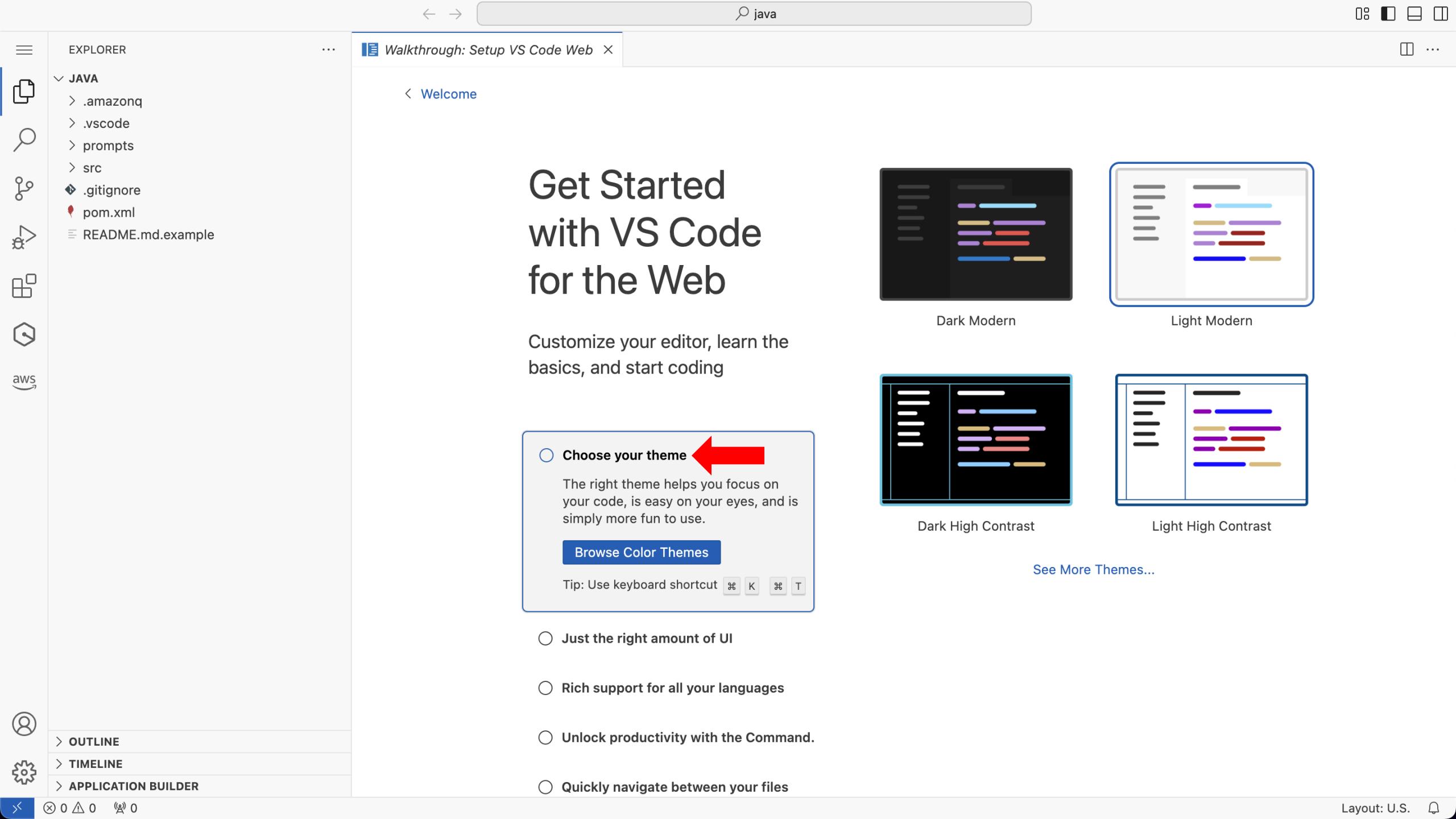
Please log in below. Check the config file at `/home/participant/.config/code-server/config.yaml` for the password.

 PASSWORDSUBMIT

Welcome to code-server

Please log in below. Check the config file at `/home/participant/.config/code-server/config.yaml` for the password.

.....|SUBMIT



← → 🔍 java

EXPLORER ... Walkthrough: Setup VS Code Web X

...

JAVA .amazonq .vscode prompts src target .gitignore pom.xml README.md.example

Get Started with VS Code for the Web

Customize your editor, learn the basics, and start coding

Choose your theme

The right theme helps you focus on your code, is easy on your eyes, and is simply more fun to use.

Browse Color Themes

Tip: Use keyboard shortcut ⌘ K ⌘ T

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

participant:/Workshop/java\$

Dark Modern Light Modern

Dark High Contrast Light High Contrast

See More Themes...

Updating Amazon Q plugin

AWS Toolkit collects anonymous usage metrics to improve the product. You can opt-out in settings.

Source: AWS Toolkit Settings OK

Amazon Q collects anonymous usage metrics to improve the product. You can opt-out in settings.

Source: Amazon Q Settings OK

OUTLINE TIMELINE APPLICATION BUILDER JAVA PROJECTS MAVEN

main ↗ 0 △ 14 ⓘ 3 ⌘ 0 AWS X Amazon Q Java: Searching... - 0% Layout: U.S. ⚡

← → 🔍 java

AMAZON Q: LOGIN ... Walkthrough: Setup VS Code Web X

Get Started with VS Code for the Web

Customize your editor, learn the basics, and start coding

○ Unlock productivity with the Command.

○ Quickly navigate between your files

✓ Mark Done ← Next Section →

Dark Modern Light Modern

Dark High Contrast Light High Contrast

See More Themes...

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

participant:/Workshop/java\$

Updating 'AmazonQ-Workspace' language server

Start using Amazon Q

Source: Amazon Q Sign In

main ↕ 0 △ 14 ⓘ 3 ⌂ 0 AWS X Amazon Q Java: Ready Activating Extensions... Layout: U.S.

AWS icon

Amazon Q icon

Terminal icon

Output icon

Debug Console icon

Code Reference Log icon

PROBLEMS icon

PORTS icon

TERMINAL icon (underlined)

OUTPUT icon

DEBUG CONSOLE icon

CODE REFERENCE LOG icon

Sign In button

The screenshot shows the Amazon Q: LOGIN interface. A red arrow points to the 'Welcome' tab in the top navigation bar.

AMAZON Q: LOGIN

Welcome X

Start

- New File...
- Open File...
- Clone Git Repository...

Recent

You have no recent folders, [open a folder](#) to start.

Choose a sign-in option:

- Use for Free**
with Builder ID, a personal profile from AWS
- Use with Pro license**

Continue

PROBLEMS 17 **OUTPUT** **DEBUG CONSOLE** **TERMINAL** **PORTS** **CODE REFERENCE LOG**

participant:/Workshop/java\$

Next Up

Deploy code-server for your team

Provision software development environments on your infrastructure with Coder.

Coder is a self-service portal which provisions via Terraform —Linux, macOS, Windows, x86, ARM, and, of course, Kubernetes based infrastructure.

[Get started →](#)

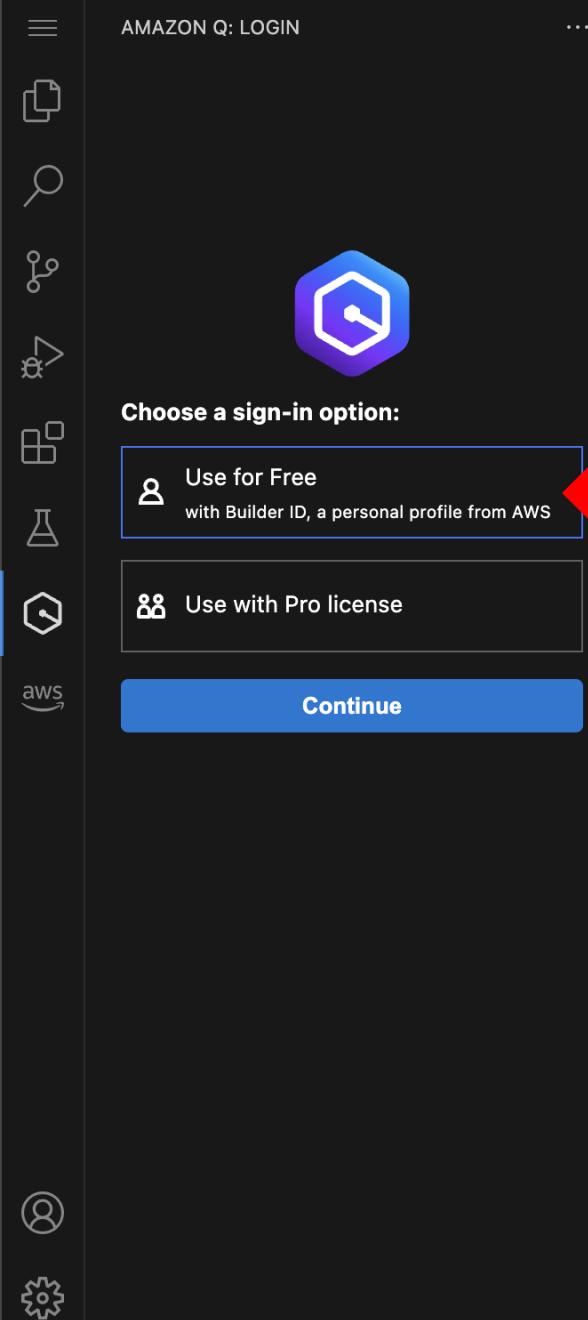
Templates

Name	Used By
Kubernetes	128 developers
Windows	21 developers

Walkthroughs

Meet Amazon Q New

Layout: U.S.



AMAZON Q: LOGIN

Choose a sign-in option:

- Use for Free**
with Builder ID, a personal profile from AWS
- Use with Pro license

Continue

Show All Commands ⌘ P
Go to File ⌘ P
Open Settings ⌘ ,
Find in Files ⌘ F

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

participant:/Workshop/java\$

AMAZON Q: LOGIN

...



Choose a sign-in option:

Use for Free

with Builder ID, a personal profile from AWS

Use with Pro license

Continue

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

participant:/Workshop/java\$

Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F

bash + - ×





AMAZON Q: LOGIN

...



Authenticating in browser...

Cancel



Confirm Code for AWS Builder ID

Confirm this code in the browser: SMMV-DPTH

Cancel

Proceed To Browser

X

P

bash + - X

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

participant:/Workshop/java\$

Create AWS Builder ID

AWS Builder ID is a new personal profile for builders. [Learn more](#)

- ✓ Get started for free
- ✓ Complement your existing AWS accounts
- ✓ Secure your login with optional MFA

Your email address

Next

OR

Already have AWS Builder ID? [Sign in](#)

By continuing and using an AWS Builder ID, you agree to the [AWS Customer Agreement](#) ("Agreement"), [AWS Service Terms](#), [AWS Privacy Notice](#), and [AWS Acceptable Use Policy](#). Your AWS Builder ID is considered an AWS account for the purposes of the Agreement. This site uses essential cookies. See our [Cookie Notice](#) for more information.





Sign in with AWS Builder ID

AWS Builder ID is a new personal profile for builders. [Learn more](#)

- ✓ Get started for free
- ✓ Complement your existing AWS accounts
- ✓ Secure your login with optional MFA

Your email address

Next

OR

Create your AWS Builder ID

By continuing and using an AWS Builder ID, you agree to the [AWS Customer Agreement](#) ("Agreement"), [AWS Service Terms](#), [AWS Privacy Notice](#), and [AWS Acceptable Use Policy](#). Your AWS Builder ID is considered an AWS account for the purposes of the Agreement. This site uses essential cookies. See our [Cookie Notice](#) for more information.





Sign in with AWS Builder ID

AWS Builder ID is a new personal profile for builders. [Learn more](#)

- ✓ Get started for free
- ✓ Complement your existing AWS accounts
- ✓ Secure your login with optional MFA

Your email address

Next

OR

Create your AWS Builder ID

By continuing and using an AWS Builder ID, you agree to the [AWS Customer Agreement](#) ("Agreement"), [AWS Service Terms](#), [AWS Privacy Notice](#), and [AWS Acceptable Use Policy](#). Your AWS Builder ID is considered an AWS account for the purposes of the Agreement. This site uses essential cookies. See our [Cookie Notice](#) for more information.





Enter your password

Username:

michael_tw_lin@msn.com (not you?)

Password

.....

Show password

[Forgot password](#)

Verify that you're a real person. Enter the characters from the image below.



6mctb2

Sign in

Cancel

This is a trusted device. [Learn more](#)





Additional verification required

We sent an email with a verification code to
michael_tw_lin@msn.com ([not you?](#))

Enter it below to confirm your identity.

Verification code

Verify

[Resend email with a code](#)





Additional verification required

We sent an email with a verification code to
michael_tw_lin@msn.com (not you?)

Enter it below to confirm your identity.

Verification code

A text input field containing the verification code "894428".An orange rectangular button labeled "Verify" in white text.

[Resend email with a code](#)





Authorization requested

An application or service has requested access to your AWS account(s) and resources.



Confirm this code matches the one given to you.

SMMV-DPTH

If you did not initiate this request or your codes do not match, cancel this request.

Confirm and continue

Cancel





Allow AWS IDE Extensions for VSCode to access your data?

By choosing **Allow access**, you agree to allow **AWS IDE Extensions for VSCode** to access the following:



Amazon Q Developer
[Show details](#)

[Deny access](#)

[Allow access](#)





✓ **Request approved**

AWS IDE Extensions for VSCode can now
access your data in Amazon Q Developer.

You can close this window.



← → 🔍 java

AMAZON Q ...

> CODE ISSUES

CHAT ...

Chat X +

NEW FEATURE X

An interactive, agentic coding experience

Amazon Q can now help you write, modify, and maintain code by combining the power of natural language understanding with the ability to take actions on your behalf such as directly making code changes, modifying files, and running commands.

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#). Amazon Q may retain chats to provide and maintain the service. For information on the AWS Regions where Amazon Q may perform inference, see [the documentation](#).

✓ Acknowledge 

Ask a question. Use @ to add context, / for quick actions

CODE ISSUES

CHAT

NEW FEATURE

aws

Meet Amazon Q

Your generative AI-powered assistant across the software development lifecycle.

Get inline code suggestions

Amazon Q suggests code as you type based on your open files. Accepted suggestions from licensed code will go into the Code Reference Log.

Try Example

TIP: Invoke manually with opt/alt + c

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

participant:/Workshop/java\$

// function to return all of the object keys in an S3 bucket
export const getKeys = async (bucketName, s3Client) => {
 const params = {
 Bucket: bucketName
 };
 const keys = [];
 let isTruncated = true;
 let continuationToken;
 while (isTruncated) {
 const response = await s3Client.listObjectsV2(params);
 isTruncated = response.IsTruncated;
 params.ContinuationToken = response.ContinuationToken;
 response.Contents.forEach(obj => keys.push(obj.Key));
 }
 return keys;
}

< 1/5 > Accept Tab ...

bash + - ×

AmazonQ: Successfully connected to AWS Builder ID

main ↕ 0 △ 14 i 3 ⌂ 0 ✓ AWS: ec2:instance ▷ Amazon Q ⌂ Java: Ready Layout: U.S. 🔍

Q Developer CLI



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Amazon Q Developer Workshop -
Building the Q-Words App

Introduction

▼ Setup and Configuration

▼ Running the Workshop at an
AWS Guided Event

Configure your IDE

▼ Running the Workshop in your
own AWS Account

Configure your IDE

Amazon Q Authentication

Setup

(Optional) Additional Languages

► Code Comprehension and
Summarization

► Code Debugging, Testing, and
Optimization

► Natural Language to New Feature

► Application Modernization and
Transformation

▼ AWS account access

[Open AWS console](#)
(us-east-1)

[Get AWS CLI credentials](#)

► Content preferences

Exit event

Part 2: Login to Amazon Q Developer for the Command Line

1. In your terminal, run:

```
1 q login
```

Copied!



2. Select **Use for Free with Builder ID** and press **Enter**

3. Open the provided URL in your browser:

- Select **Confirm and continue**
- Select **Allow access**
- After seeing "Request approved", return to your terminal

4. Test your setup with:

```
1 q chat "Hello, Amazon Q!" --no-interactive
```

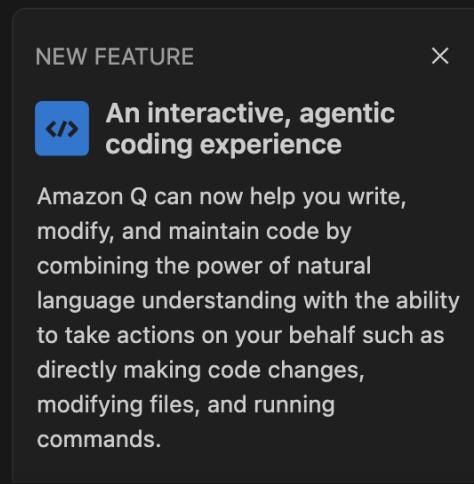
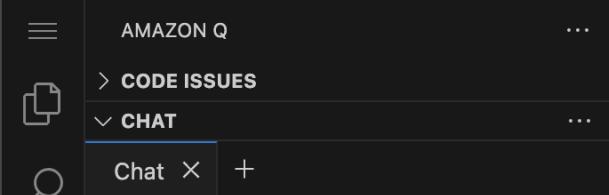


Part 3: Enable Workspace Context

To make Amazon Q aware of your entire codebase:

1. Click on 'Amazon Q' in the status bar
2. Select '**Open Settings**'
3. Select the check-box under **Amazon Q: Workspace Index** to enable workspace indexing





Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#). Amazon Q may retain chats to provide and maintain the service. For information on the AWS Regions where Amazon Q may perform inference, see [the documentation](#).

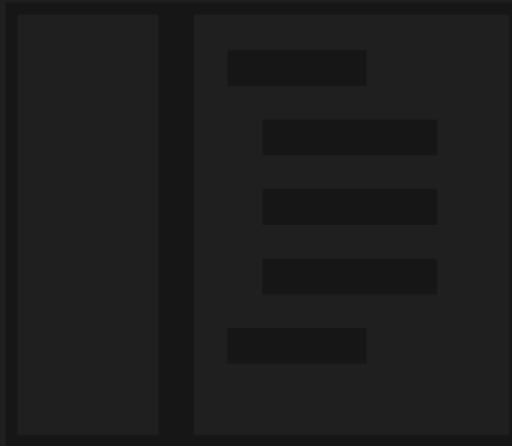
✓ Acknowledge

Ask a question. Use @ to add context, / for quick actions



@ ←

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

○ participant:/Workshop/java\$

bash + ⊞ ⊖ ⌛ ⌂ ⌁ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌊ ⌋

ⓘ AmazonQ: Successfully connected to AWS Builder ID

AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Chat X +

NEW FEATURE X

An interactive, agentic coding experience

Amazon Q can now help you write, modify, and maintain code by combining the power of natural language understanding with the ability to take actions on your behalf such as directly making code changes, modifying files, and running commands.

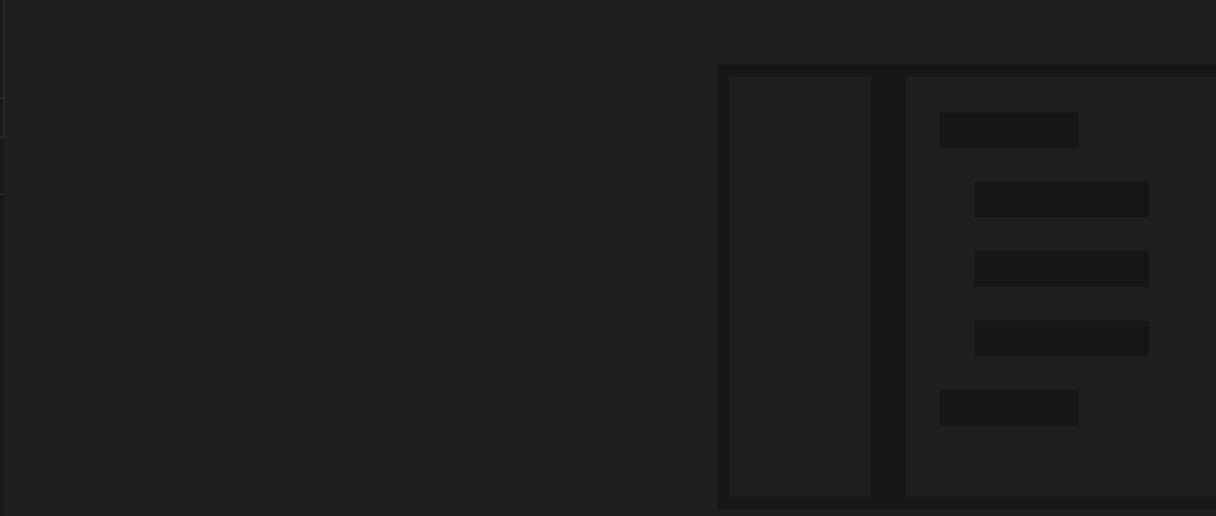
Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#). Amazon Q may retain chats to provide and maintain the service. For information on the AWS Regions where Amazon Q may perform inference, see [the documentation](#).

✓ Acknowledge

Ask a question. Use @ to add context, / for quick actions

@ ←

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ ⌘ F

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

○ participant:/Workshop/java\$

bash + □ ✎ ⌛ ⌂ ⌄

- bash
- tmux
- JavaScript Debug Terminal
- Split Terminal >
- Configure Terminal Settings**
- Select Default Profile
- Run Task...
- Configure Tasks...

This screenshot shows the AWS Toolkit for Visual Studio Code interface. The top navigation bar includes icons for file operations (New, Open, Save, etc.) and a search bar with the query "java". The left sidebar features the "AMAZON Q" extension, with sections for "CODE ISSUES" and "CHAT". A "NEW FEATURE" callout highlights the "An interactive, agentic coding experience" provided by Amazon Q, mentioning its ability to help write, modify, and maintain code using natural language understanding and AI actions. Below this, a note about generative AI and the AWS Responsible AI Policy is displayed, along with a link to documentation and an "Acknowledge" button.

The main workspace shows a terminal window with the command "participant:/Workshop/java\$". The bottom status bar displays various icons and the message "AWS: ec2:instance" and "Amazon Q Java: Ready".

A settings search results panel is open, showing 51 results for "font". The results are categorized under "User", "Remote [d2v4mt5uwdherk.cloudfront.net]", and "Workspace". The "User" tab is selected, displaying settings like "Editor: Font Size" (set to 12), "Editor: Font Family" (set to Menlo, Monaco, 'Courier New', monospace), "Editor: Code Lens Font Family" (set to Menlo, Monaco, 'Courier New', monospace), and "Editor: Code Lens Font Size" (set to 12).

The screenshot shows the AWS Toolkit for VS Code interface. The top navigation bar includes icons for back, forward, search (with 'java' typed), and system controls. The left sidebar has sections for 'AMAZON Q', 'CODE ISSUES' (expanded to show 'CHAT'), and 'Chat'. A prominent 'NEW FEATURE' box highlights 'An interactive, agentic coding experience' with a description of Amazon Q's capabilities. Below it, a note about generative AI and the AWS Responsible AI Policy is shown, with a 'Acknowledge' button. The main area displays the 'Settings' sidebar with a search bar for 'font'. The results are filtered by 'User' (selected) and 'Remote [d2v4mt5uwdherk.cloudfront.net]' and 'Workspace'. The 'Commonly Used (2)' section lists 'Text Editor (17)' which is expanded to show 'Font (5)' (selected), 'Minimap (2)', 'Suggestions (2)', 'Workbench (2)', 'Features (26)', 'Extensions (6)', and 'CSS Language F... (3)'. The 'Font' section contains three settings: 'Editor: Font Size' (set to 12), 'Editor: Font Family' (set to 'Menlo, Monaco, \'Courier New\', monospace'), and 'Editor: Font Ligatures' (with a link to 'Edit in settings.json'). The bottom of the screen shows the terminal tab with the command 'participant:/Workshop/java\$' and the status bar with various icons and text.

AMAZON Q

> CODE ISSUES

CHAT

Chat X +

NEW FEATURE

An interactive, agentic coding experience

Amazon Q can now help you write, modify, and maintain code by combining the power of natural language understanding with the ability to take actions on your behalf such as directly making code changes, modifying files, and running commands.

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#). Amazon Q may retain chats to provide and maintain the service. For information on the AWS Regions where Amazon Q may perform inference, see [the documentation](#).

✓ Acknowledge

Ask a question. Use @ to add context, / for quick actions

AMAZON Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

Settings X

font

User Remote [d2v4mt5uwdherk.cloudfront.net] Workspace

Commonly Used (2)

Text Editor (17)

Font (5)

Minimap (2)

Suggestions (2)

Workbench (2)

Appearance (1)

Screencast Mode (1)

Features (26)

Debug (3)

Source Control (2)

Terminal (12)

Notebook (5)

Chat (4)

Extensions (6)

CSS Language F... (3)

Editor: Font Size

Controls the font size in pixels.

12

Editor: Font Family

Controls the font family.

Menlo, Monaco, 'Courier New', monospace

Editor: Font Ligatures

Configures font ligatures or font features. Can be either a boolean to enable/disable ligatures or a string for the value of the CSS 'font-feature-settings' property.

Edit in settings.json

Editor: Font Variations

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

bash + ⊞ ⊖ ⊗ ^ ×

participant:/Workshop/java\$

main ⊞ ⊖ 0 △ 14 ⊖ 3 ⌂ 0 ✓ AWS: ec2:instance ▷ Amazon Q ⌂ Java: Ready

Layout: U.S. ⌂

This screenshot shows the Amazon Q extension integrated into the Visual Studio Code interface. The top navigation bar includes icons for file, edit, and search, along with a search bar containing 'java'. The left sidebar features the 'AMAZON Q' extension icon, a 'CODE ISSUES' section, and a 'CHAT' section with a 'Chat' tab currently selected. A large callout box highlights the 'NEW FEATURE' section, which describes the interactive, agentic coding experience provided by Amazon Q.

The main content area displays the 'Settings' page for the 'font' category. The 'User' scope is selected, showing 51 settings found. The 'Editor: Font Size' setting is set to 15 pixels. The 'Editor: Font Family' setting is set to 'Menlo, Monaco, 'Courier New', monospace'. The 'Editor: Font Ligatures' setting is configured to enable ligatures. The 'Editor: Font Variations' setting is listed but not yet configured.

At the bottom, a terminal window shows the command 'participant:/Workshop/java\$'.

Bottom navigation bar: PROBLEMS (17), OUTPUT, DEBUG CONSOLE, TERMINAL (selected), PORTS, CODE REFERENCE LOG.

Bottom status bar: Layout: U.S., AWS: ec2:instance, Amazon Q, Java: Ready.

← → 🔍 java

AMAZON Q ... ➤ CODE ISSUES CHAT Chat X +

NEW FEATURE An interactive, agentic coding experience

Amazon Q can now help you write, modify, and maintain code by combining the power of natural language understanding with the ability to take actions on your behalf such as directly making code changes, modifying files, and running commands.

Amazon Q Developer uses generative AI. You may need to verify responses. See the AWS Responsible AI Policy. Amazon Q may retain chats to provide and maintain the service. For information on the AWS Regions where Amazon Q may perform inference, see the documentation.

✓ Acknowledge

Ask a question. Use @ to add context, / for quick actions

Amazon Q Developer uses generative AI. You may need to verify responses. See the AWS Responsible AI Policy.

Settings X font

User Remote [d2v4mt5uwdherk.cloudfront.net] Workspace

Text Editor (17)

- Font (5)
- Minimap (2)
- Suggestions (2)

Workbench (2)

- Appearance (1)
- Screencast Mode (1)

Features (26)

- Debug (3)
- Terminal (2)
- Terminal (12)
- Notebook (5)
- Chat (4)

Extensions (6)

- CSS Language F... (3)
- CSS (1)

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

bash + ×

java

51 Settings Found

Font Family

Controls the font family of the terminal. Defaults to Editor: Font Family's value.

Font Ligatures: Enabled

Controls whether font ligatures are enabled in the terminal. Ligatures will only work if the configured Terminal > Integrated: Font Family supports them.

Font Ligatures: Fallback Ligatures

When Terminal > Integrated: Gpu Acceleration is enabled and the particular Terminal > Integrated: Font Family cannot be parsed, this is the set of character sequences that will always be drawn together. This allows the use of a fixed set of ligatures even when the font isn't supported.

Edit in settings.json

Layout: U.S.

The screenshot shows the Visual Studio Code (VS Code) interface with the following details:

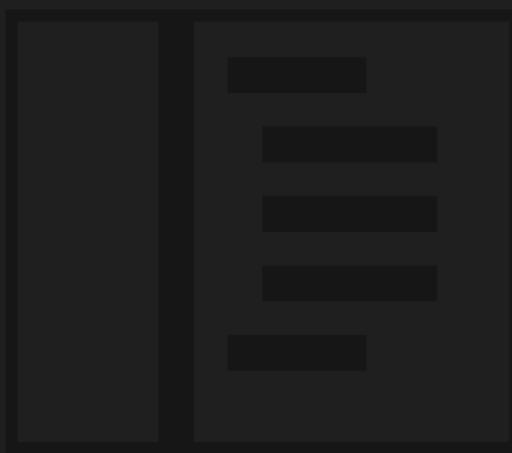
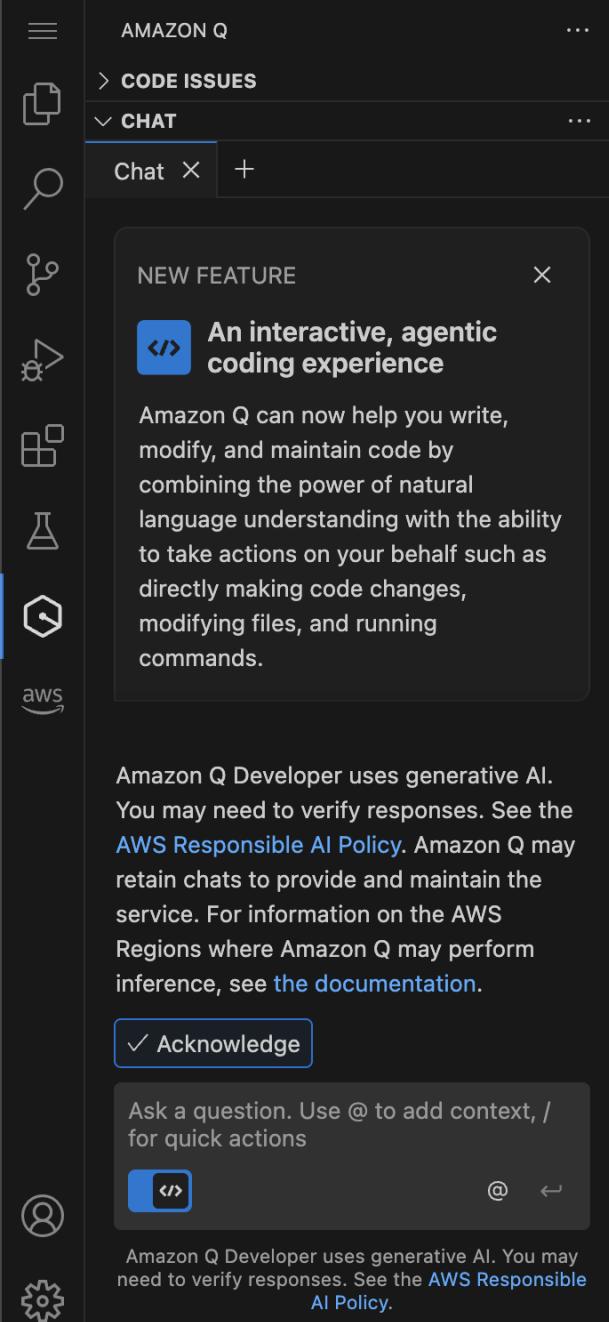
- Top Bar:** Includes icons for file operations (New, Open, Save, etc.), a search bar with the text "java", and system status icons.
- Sidebar:** On the left, there's a vertical toolbar with icons for File, Edit, Find, Search, Terminal, and AWS services. Below it, the "AMAZON Q" extension is active, displaying sections for "CODE ISSUES" (with a "CHAT" sub-section), "NEW FEATURE" (describing AI code editing), and "AWS Responsible AI Policy" (warning about AI inference).
- Central Area:** The main workspace shows a "Settings" sidebar open under the "User" tab. The search term "font" is entered. The results list includes categories like "Text Editor", "Workbench", "Features", "Extensions", and "Terminal".
 - Text Editor (17):** Includes "Font" (5), "Minimap" (2), and "Suggestions" (2).
 - Workbench (2):** Includes "Appearance" (1) and "Screencast Mode" (1).
 - Features (26):** Includes "Debug" (3), "Source Control" (2), and "Terminal" (12).
 - Extensions (6):** Includes "CSS Language F..." (3) and "CSS" (1).
- Terminal:** Shows the command "calt" being typed into the integrated terminal, with suggestions "off", "ss03", "on", and "dlig" appearing below it.
- Bottom Bar:** Includes tabs for PROBLEMS (17), OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), PORTS, and CODE REFERENCE LOG. It also shows the current path "participant:/Workshop/java\$".
- Bottom Status Bar:** Displays navigation icons (File, Open, Save, etc.), the status "AWS: ec2:instance", the status "Amazon Q: Java: Ready", and the text "Layout: U.S."

This screenshot shows the Visual Studio Code interface with the Amazon Q extension installed. The left sidebar features the Amazon Q icon, a 'Code Issues' section, and a 'Chat' section. A prominent 'NEW FEATURE' box highlights 'An interactive, agentic coding experience' where Amazon Q can help write, modify, and maintain code. Below this, a note about generative AI and the AWS Responsible AI Policy is displayed, with a 'Acknowledge' button.

The main area shows the 'Settings' page with a search bar for 'java'. The results list 'font' under the 'User' tab. The 'Text Editor' section includes 'Font' (5), 'Minimap' (2), and 'Suggestions' (2). The 'Workbench' section includes 'Appearance' (1) and 'Screencast Mode' (1). The 'Features' section includes 'Debug' (3), 'Source Control' (2), and 'Terminal' (12). The 'Extensions' section includes 'CSS Language F...' (3) and 'CSS' (1).

A tooltip for 'Terminal > Integrated: Font Size' indicates it controls the font size in pixels of the terminal, currently set to 15. The 'Font Weight' setting is also mentioned, allowing 'normal' or 'bold' text or numbers between 1 and 1000. A terminal window at the bottom shows the command 'participant:/Workshop/java\$'.

At the bottom, the status bar shows 'main' and 'AWS: ec2:instance' along with other standard VS Code icons.



Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#). Amazon Q may retain chats to provide and maintain the service. For information on the AWS Regions where Amazon Q may perform inference, see [the documentation](#).

✓ Acknowledge

Ask a question. Use @ to add context, / for quick actions



@ ↵

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

d2v4mt5uwdhherk.cloudfront. X

net wants to

See text and images copied to the clipboard

Block Allow

An interactive, agentic coding experience

Amazon Q can now help you write, modify, and maintain code by combining the power of natural language understanding with the ability to take actions on your behalf such as directly making code changes, modifying files, and running commands.

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#). Amazon Q may retain chats to provide and maintain the service. For information on the AWS Regions where Amazon Q may perform inference, see [the documentation](#).

✓ Acknowledge

Ask a question. Use @ to add context, / for quick actions

@ ↵

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

java

Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F

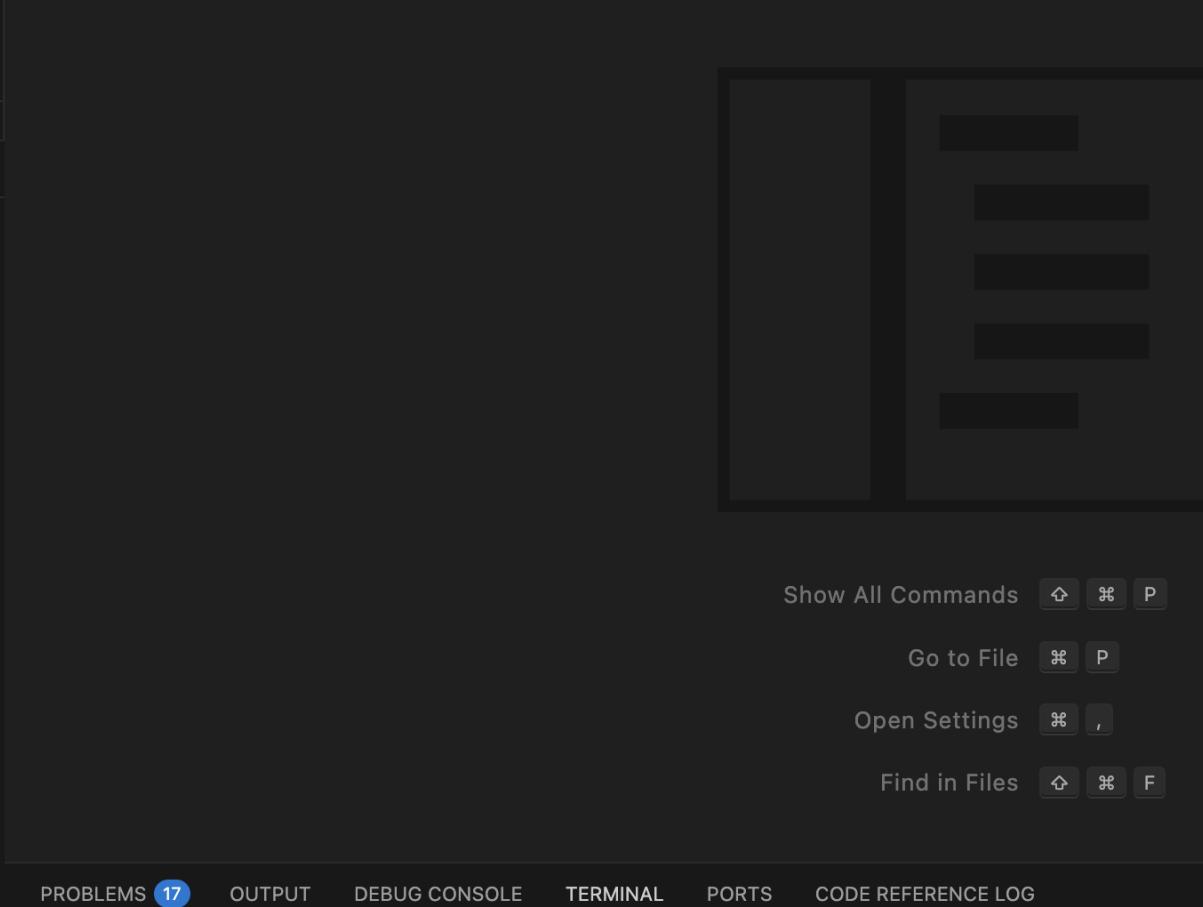
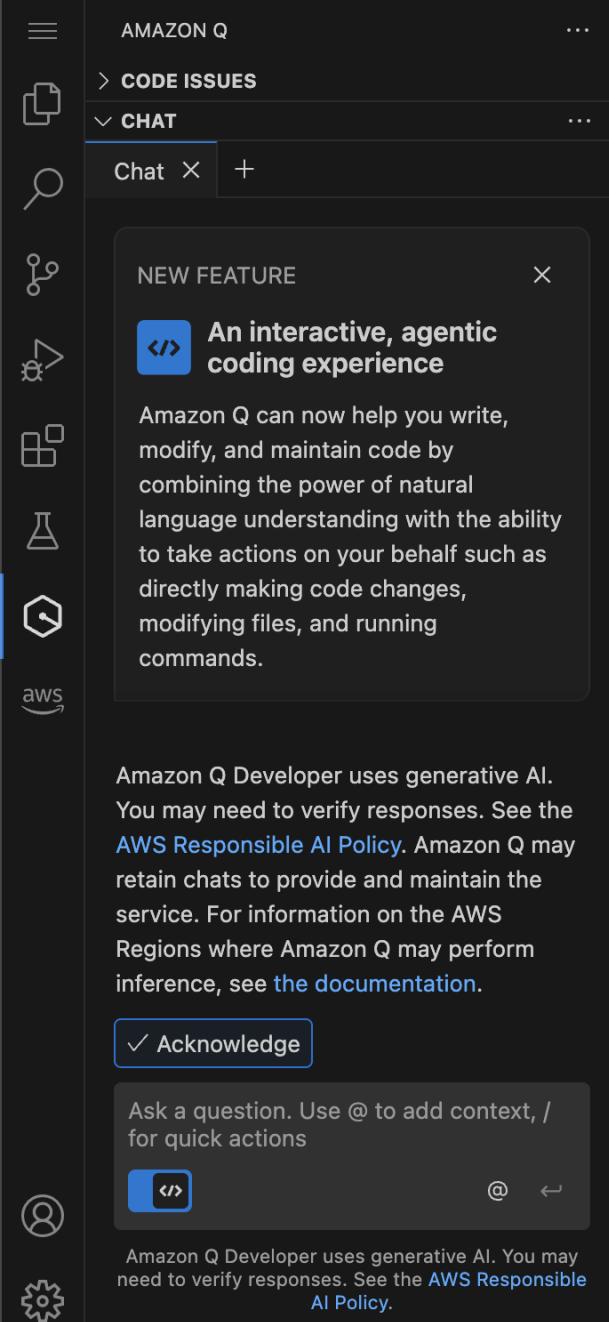
PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

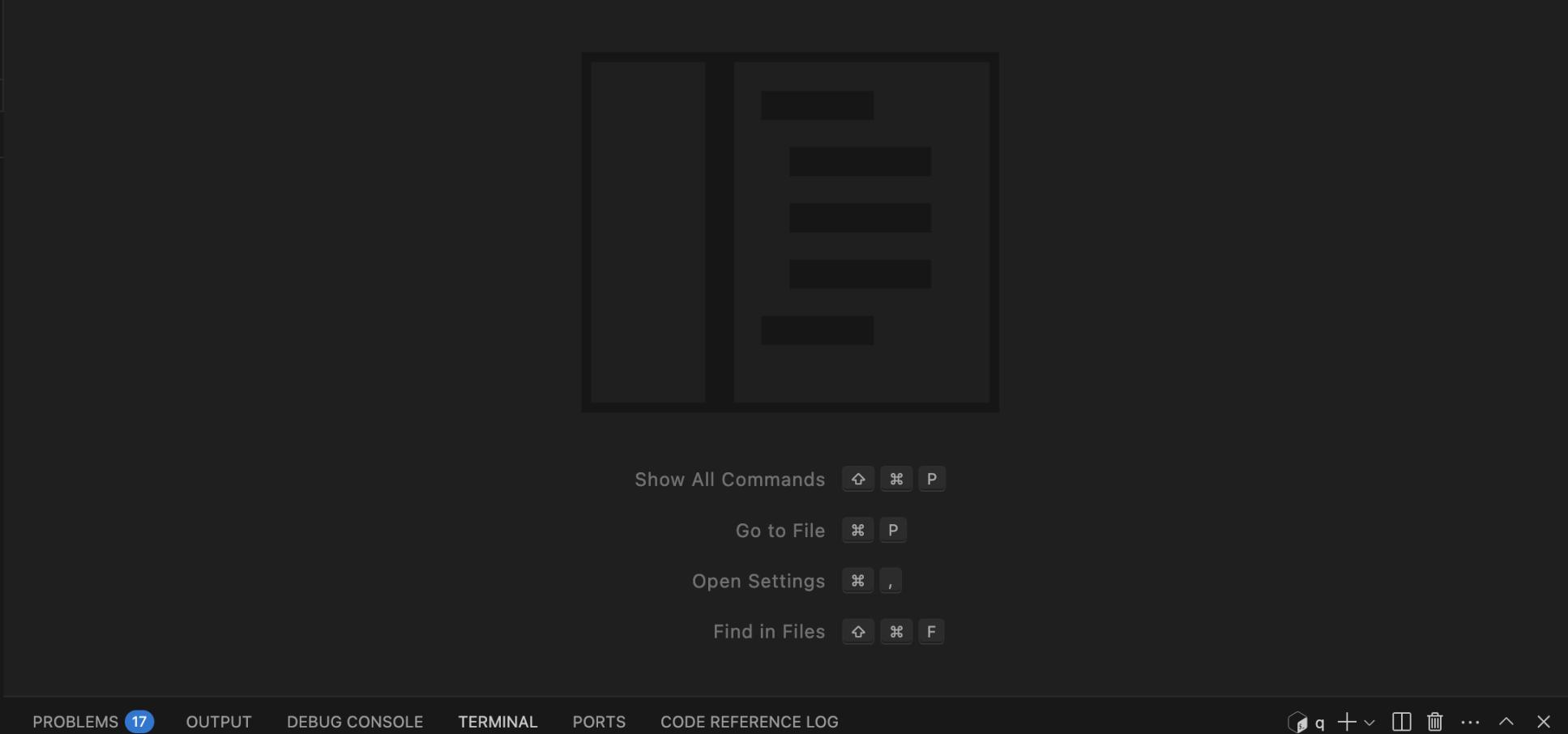
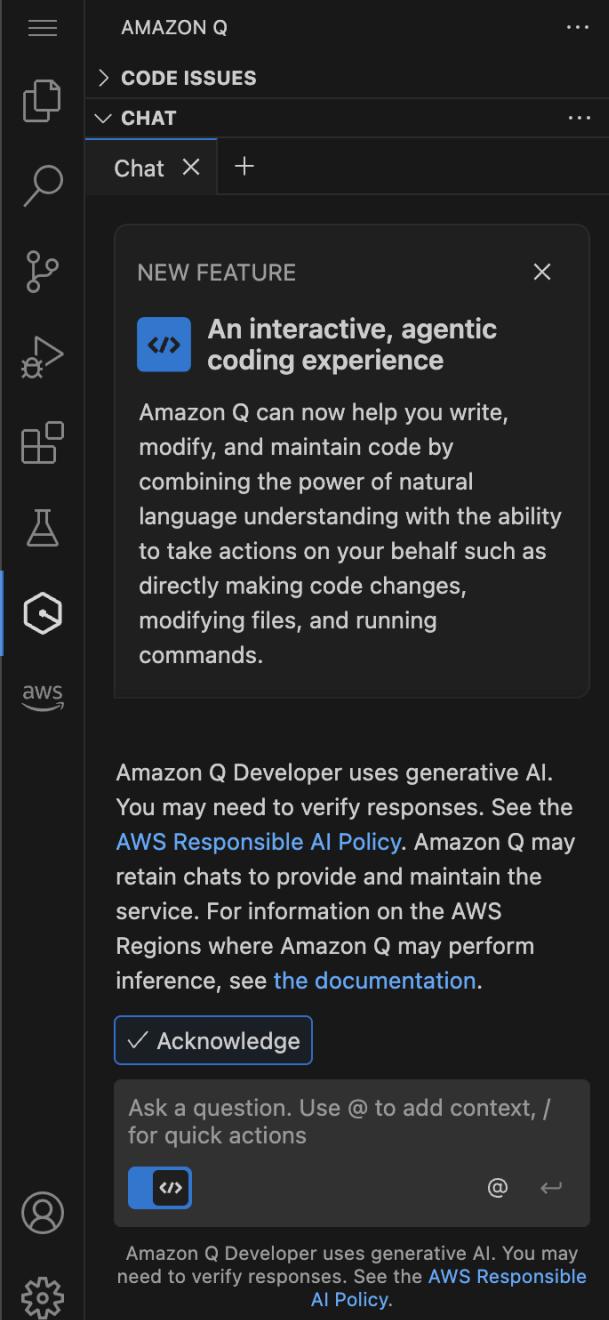
participant:/Workshop/java\$

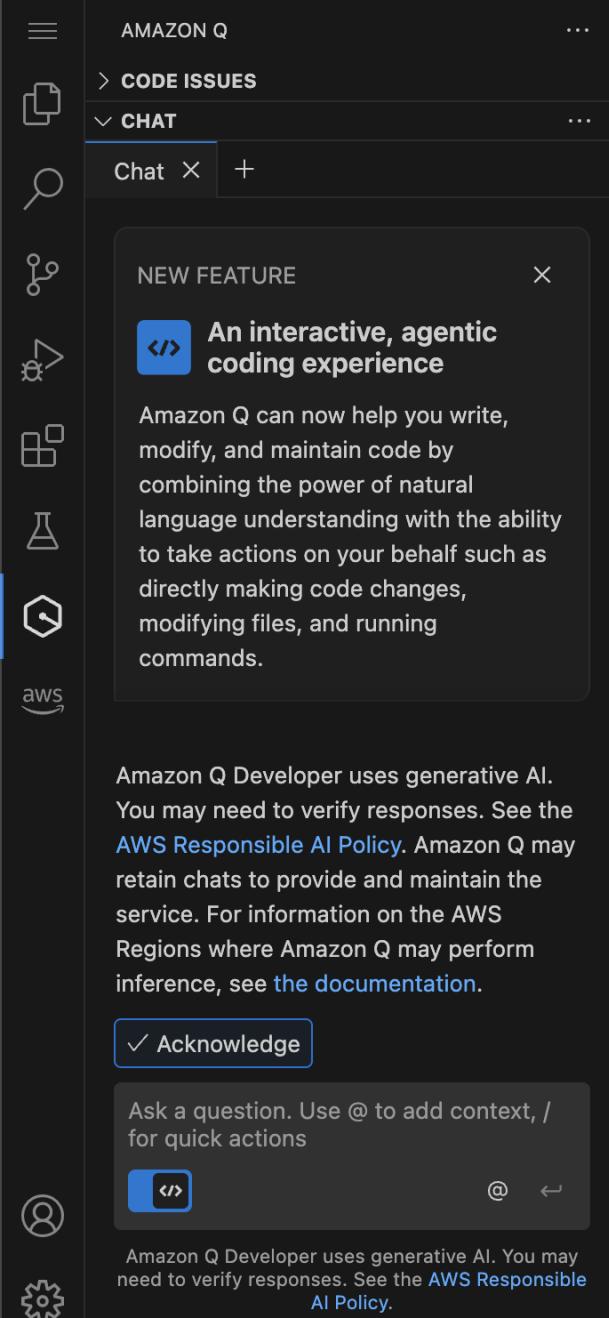
bash + ⊞ ⊖ ⌛ ⌁ ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋

x main ⊞ ⊖ 0 △ 14 ⊕ 3 ⌂ 0 ✓ AWS: ec2:instance D Amazon Q D Java: Ready

Layout: U.S. 🔍







The terminal tab in VS Code shows the following interaction:

```
PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS 1 CODE REFERENCE LOG
○ participant:/Workshop/java$ q login
✓ Select login method · Use for Free with Builder ID

Confirm the following code in the browser
Code: CLFV-FGSD

Open this URL: https://view.awsapps.com/start/#/device?user_code=CLFV-FGSD ←
```

A red arrow points to the URL in the terminal output, indicating the next step for logging in.

AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Chat X +

NEW FEATURE X

An interactive, agentic coding experience

Amazon Q can now help you write, modify, and maintain code by combining the power of natural language understanding with the ability to take actions on your behalf such as directly making code changes, modifying files, and running commands.

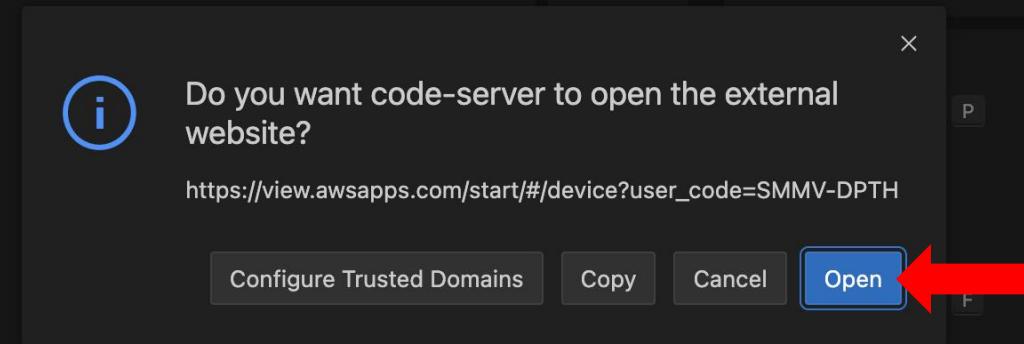
Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#). Amazon Q may retain chats to provide and maintain the service. For information on the AWS Regions where Amazon Q may perform inference, see [the documentation](#).

✓ Acknowledge

Ask a question. Use @ to add context, / for quick actions

Code editor icon @ ↶

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS 1 CODE REFERENCE LOG

o participant:/Workshop/java\$ q login
✓ Select login method · Use for Free with Builder ID

Confirm the following code in the browser
Code: CLFV-FGSD

Open this URL: https://view.awsapps.com/start/#/device?user_code=CLFV-FGSD
Logging in...

main ↹ 0 △ 14 ⌂ 3 ⌂ 1 ✓ AWS: ec2:instance ▷ Amazon Q ⌂ Java: Ready Layout: U.S. ⌂



Authorization requested

An application or service has requested access to your AWS account(s) and resources.



Confirm this code matches the one given to you.

CLFV-FGSD

If you did not initiate this request or your codes do not match, cancel this request.

Confirm and continue

Cancel





Allow Amazon Q Developer for command line to access your data?

By choosing **Allow access**, you agree to allow **Amazon Q Developer for command line** to access the following:



Amazon Q Developer
[Show details](#)

[Deny access](#)

[Allow access](#)



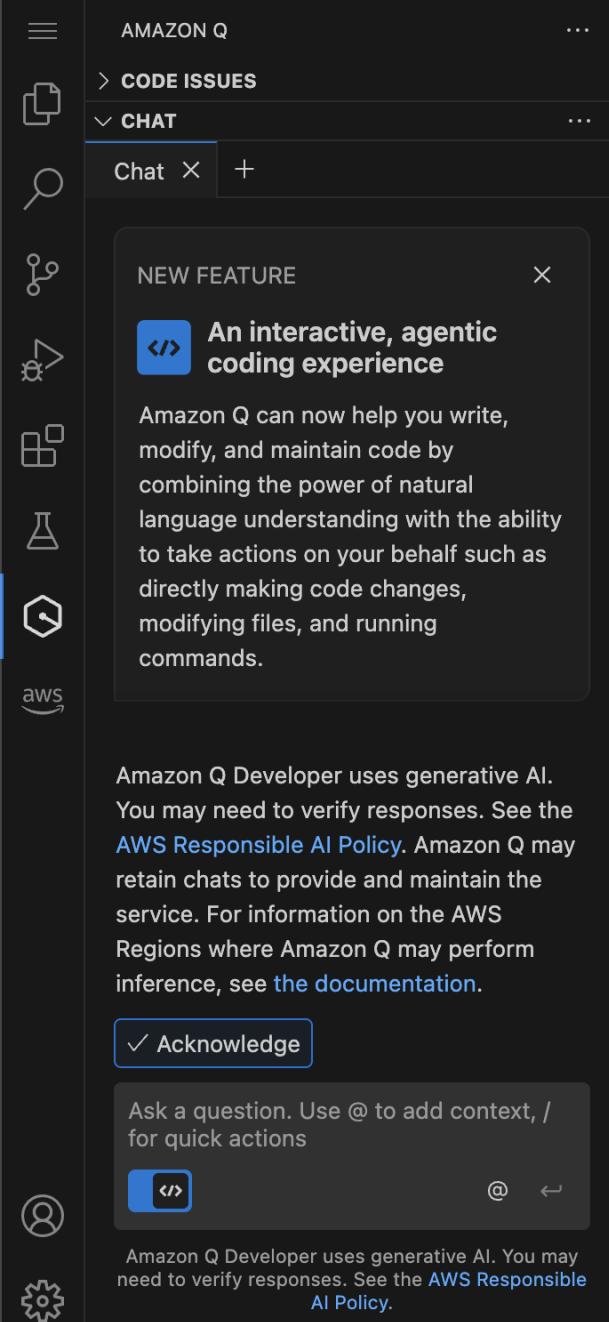


✓ **Request approved**

Amazon Q Developer for command line can
now access your data in Amazon Q Developer.

You can close this window.





Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

participant:/Workshop/java\$ q login
✓ Select login method · Use for Free with Builder ID

Confirm the following code in the browser
Code: CLFV-FGSD

Open this URL: https://view.awsapps.com/start/#/device?user_code=CLFV-FGSD
Device authorized
Logged in successfully

participant:/Workshop/java\$

main ↗ 0 △ 14 ⓘ 3 ⌘ 0 ✓ AWS: ec2:instance ▷ Amazon Q ⌘ Java: Ready Layout: U.S. ⌘

Run the App



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

(Optional) Additional Languages

▼ Code Comprehension and Summarization

Task 1: Learn about Amazon Q

Task 2: Learn About the Application Structure

Task 3: Create documentation

▼ Code Debugging, Testing, and Optimization

Task 1: Run and Play Q-Words

Task 2: Identify and Fix a Bug with Amazon Q

Task 3: Writing Unit Tests with Amazon Q Developer

Task 4: Review your code with Amazon Q Developer

► Natural Language to New Feature

► Application Modernization and Transformation

► Generative AI for Accelerating Software Development

Resource Cleanup

Wrapping Up: Next Steps and Feedback

► AWS account access

► Content preferences

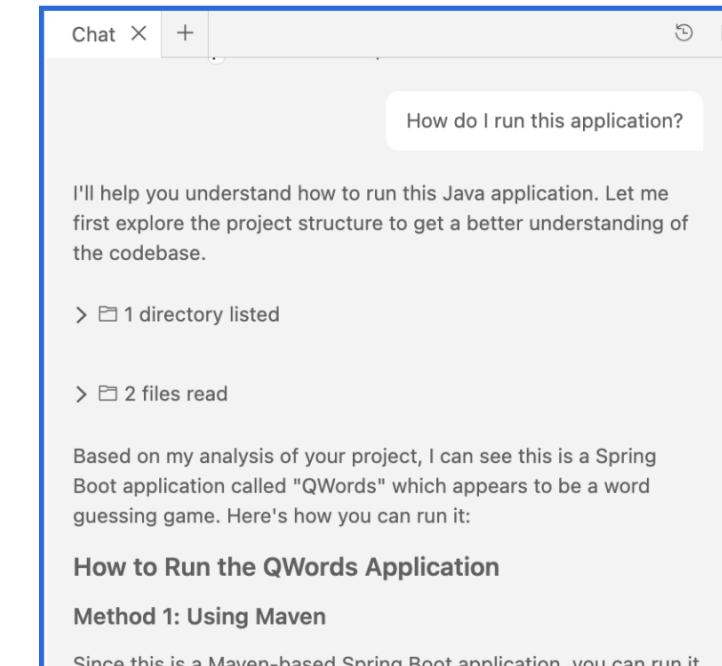
Exit event

Activity 1: Ask Amazon Q in the IDE

1. Make sure agentic coding is turned **ON** in Ama

Copied!

2. Ask Amazon Q: How do I run this application?



3. Follow the instructions provided by Amazon Q.

Reflection

- How detailed are the instructions provided by Amazon Q?
- How does Amazon Q determine the correct way to run this specific application?



AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Chat X Chat X +

NEW FEATURE X

An interactive, agentic coding experience

Amazon Q can now help you write, modify, and maintain code by combining the power of natural language understanding with the ability to take actions on your behalf such as directly making code changes, modifying files, and running commands.

Hi, I'm Amazon Q. I can answer your software development questions.
Ask me to explain, debug, or optimize your code.
You can enter / to see a list of quick actions.

Agentic coding - OFF

Agentic coding - ON

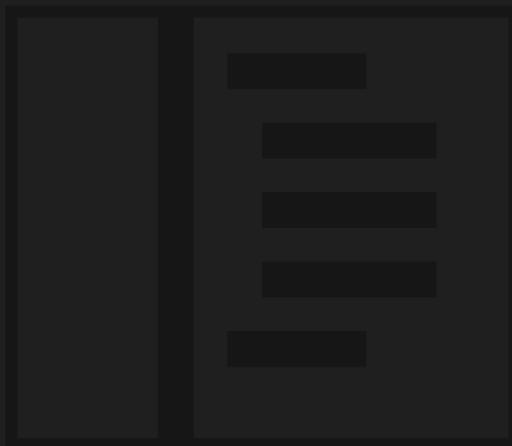
Agentic coding - OFF

Agentic coding - ON

How do I run this application? 

@ ←

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F



AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Chat X Chat X +

How do I run this application?

> Context

I'll help you understand how to run this application. Let me first explore the project structure to get a better understanding of what we're working with.

> 1 directory listed

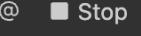
Now let's look at the pom.xml file to understand what kind of Java application this is:

> 3 files read

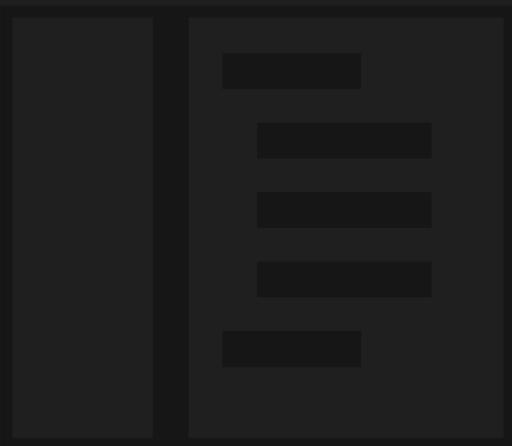
Based on my exploration of your project, I can see that Q-Words is a Spring Boot web application that implements a word-guessing game similar to Wordle. Here's how to run the application:

How to Run the Q-Words Application

Ask a question. Use @ to add context, / for quick actions

 @ 

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



Show All Commands   P

Go to File   P

Open Settings  

Find in Files   F



AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Chat X Chat X +

Running the Application

1. Open a terminal in your project directory
(/Workshop/java)
2. Build the application using Maven:

```
mvn clean install
```

 Copy

3. Run the Spring Boot application:

```
mvn spring-boot:run
```

 Copy

4. Once the application starts, open a web browser and navigate to:

```
http://localhost:8080
```

 Copy

Alternative Running Method

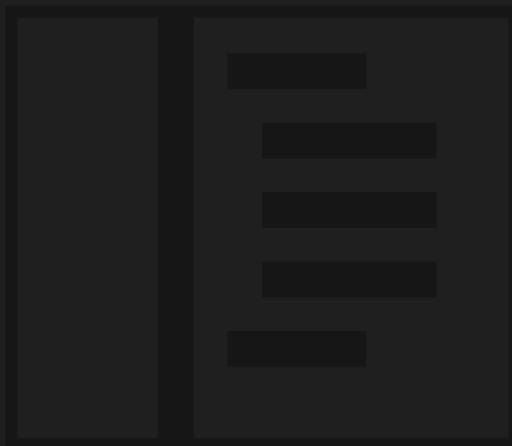
You can also run the application directly using the Spring Boot Maven plugin:

```
mvn spring-boot:run
```

Ask a question. Use @ to add context, / for quick actions

  @ ←

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



Show All Commands   P

Go to File  P

Open Settings  ,

Find in Files   F



Amazon Q Authentication Setup

(Optional) Additional Languages

▼ Code Comprehension and Summarization

Task 1: Learn about Amazon Q

Task 2: Learn About the Application Structure

Task 3: Create documentation

▼ Code Debugging, Testing, and Optimization

Task 1: Run and Play Q-Words

Task 2: Identify and Fix a Bug with Amazon Q

Task 3: Writing Unit Tests with Amazon Q Developer

Task 4: Review your code with Amazon Q Developer

▼ AWS account access

[Open AWS console
\(us-east-1\)](#) [Get AWS CLI credentials](#)

▼ Content preferences

Language

English

Exit event

Activity 2: Use Amazon Q CLI

1. Open a terminal and start an Amazon Q chat:

1 q chat



2. Ask Amazon Q CLI: Can you run this Spring Boot application for me?

```
/help all commands • ctrl + j new lines • ctrl + k fuzzy search
> can you run this Spring Boot application for me?

I'd be happy to help you run a Spring Boot application. However, I need to first check if there's a Spring Boot application in your current directory. Let me examine the files in your current directory to see what we're working with.

⌘ Using tool: fs_read (trusted)
:
• Reading directory: /Workshop/java with maximum depth of 0
:
• Completed in 0.0s

I can see this is a Java project with a pom.xml file, which suggests it's a Maven project. Let me check the pom.xml to
```

3. Amazon Q CLI will run the necessary commands to start the application.

Tip: Amazon Q CLI can execute commands directly in your terminal, making it efficient for command-line operations.

Activity 3: Run Manually (Optional)

For Java version:

```
1
2 # Set Java environment
```



AMAZON Q

> CODE ISSUES

CHAT

Q - Doc X Chat X +

Profiles

The application has two profiles:

- local : For local development
- prod : For production deployment (default)

To run with a specific profile:

```
mvn spring-boot:run -Dspring-boot.run.profiles=local
```

bash Insert at cursor Copy

Troubleshooting

If you encounter issues:

1. Verify port 8090 is available
2. Check the console logs for detailed error messages
3. Ensure Java 8 is installed: `java -version`
4. If using AWS integration, verify your AWS credentials are properly configured

The application follows an MVC architecture pattern with controllers handling HTTP requests, services managing game logic, and Thymeleaf templates rendering the user interface.

Ask a question. Use @ to add context, / for quick actions

@ ↵

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

participant:/Workshop/java\$ q chat



AMAZON Q

> CODE ISSUES

✓ CHAT

Q - Doc X Chat X +

Profiles

The application has two profiles:

- local : For local development
- prod : For production deployment (default)

To run with a specific profile:

```
mvn spring-boot:run -Dspring-boot.run.profiles=local
```

bash Insert at cursor Copy

Troubleshooting

If you encounter issues:

- Verify port 8090 is available
- Check the console logs for detailed error messages
- Ensure Java 8 is installed: `java -version`
- If using AWS integration, verify your AWS credentials are properly configured

The application follows an MVC architecture pattern with controllers handling HTTP requests, services managing game logic, and Thymeleaf templates rendering the user interface.

Ask a question. Use @ to add context, / for quick actions

@ ↵

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

participant:/Workshop/java\$ q chat



Did you know?

You can resume the last conversation from your current directory by launching with `q chat --resume`

/help all commands • `ctrl + j` new lines • `ctrl + s` fuzzy search

>



(Optional) Additional Languages

▼ Code Comprehension and Summarization

Task 1: Learn about Amazon Q

Task 2: Learn About the Application Structure

Task 3: Create documentation

▼ Code Debugging, Testing, and Optimization

Task 1: Run and Play Q-Words

Task 2: Identify and Fix a Bug with Amazon Q

Task 3: Writing Unit Tests with Amazon Q Developer

Task 4: Review your code with Amazon Q Developer

► Natural Language to New Feature

► Application Modernization and Transformation

► Generative AI for Accelerating Software Development

Resource Cleanup

Wrapping Up: Next Steps and Feedback

► AWS account access

► Content preferences

Exit event

2. Ask Amazon Q CLI: Can you run this Spring Boot application for me?

Copied!

```
/help all commands • ctrl + j new lines • ctrl + k fuzzy search
> can you run this Spring Boot application for me?

I'd be happy to help you run a Spring Boot application. However, I need to first check if there's a Spring Boot application in your current directory. Let me examine the files in your current directory to see what we're working with.

✖ Using tool: fs_read (trusted)
⋮
● Reading directory: /Workshop/java with maximum depth of 0
⋮
● Completed in 0.0s

T can see this is a Java project with a pom.xml file, which suggests it's a Maven project. Let me check the pom.xml to
```

3. Amazon Q CLI will run the necessary commands to start the application.

Tip: Amazon Q CLI can execute commands directly in your terminal, making it efficient for command-line operations.

Activity 3: Run Manually (Optional)

For Java version:

```
1
2 # Set Java environment
3 export JAVA_HOME=/usr/lib/jvm/java-1.8.0-amazon-corretto
```



```
1
2 # Build and run
3
4 mvn clean package
```



AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Chat X Chat X +

Running the Application

1. Open a terminal in your project directory
(/Workshop/java)
2. Build the application using Maven:

```
mvn clean install
```


3. Run the Spring Boot application:

```
mvn spring-boot:run
```


4. Once the application starts, open a web browser and navigate to:
<http://localhost:8080>

Alternative Running Method

You can also run the application directly using the Spring Boot Maven plugin:

```
mvn spring-boot:run
```

Ask a question. Use @ to add context, / for quick actions

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

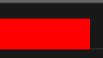
main* 0 14 3 0 ✓ AWS: ec2:instance

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

Did you know?

You can use `/editor` to edit your prompt with a vim-like experience

/help all commands • `ctrl + j` new lines • `ctrl + s` fuzzy search

> Can you run this Spring Boot application for me? 

Amazon Q Java: Ready



AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Chat X Chat X +

Running the Application

1. Open a terminal in your project directory
(/Workshop/java)
2. Build the application using Maven:

```
mvn clean install
```


3. Run the Spring Boot application:

```
mvn spring-boot:run
```


4. Once the application starts, open a web browser and navigate to:
<http://localhost:8080>

Alternative Running Method

You can also run the application directly using the Spring Boot Maven plugin:

```
mvn spring-boot:run
```

Ask a question. Use @ to add context, / for quick actions

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

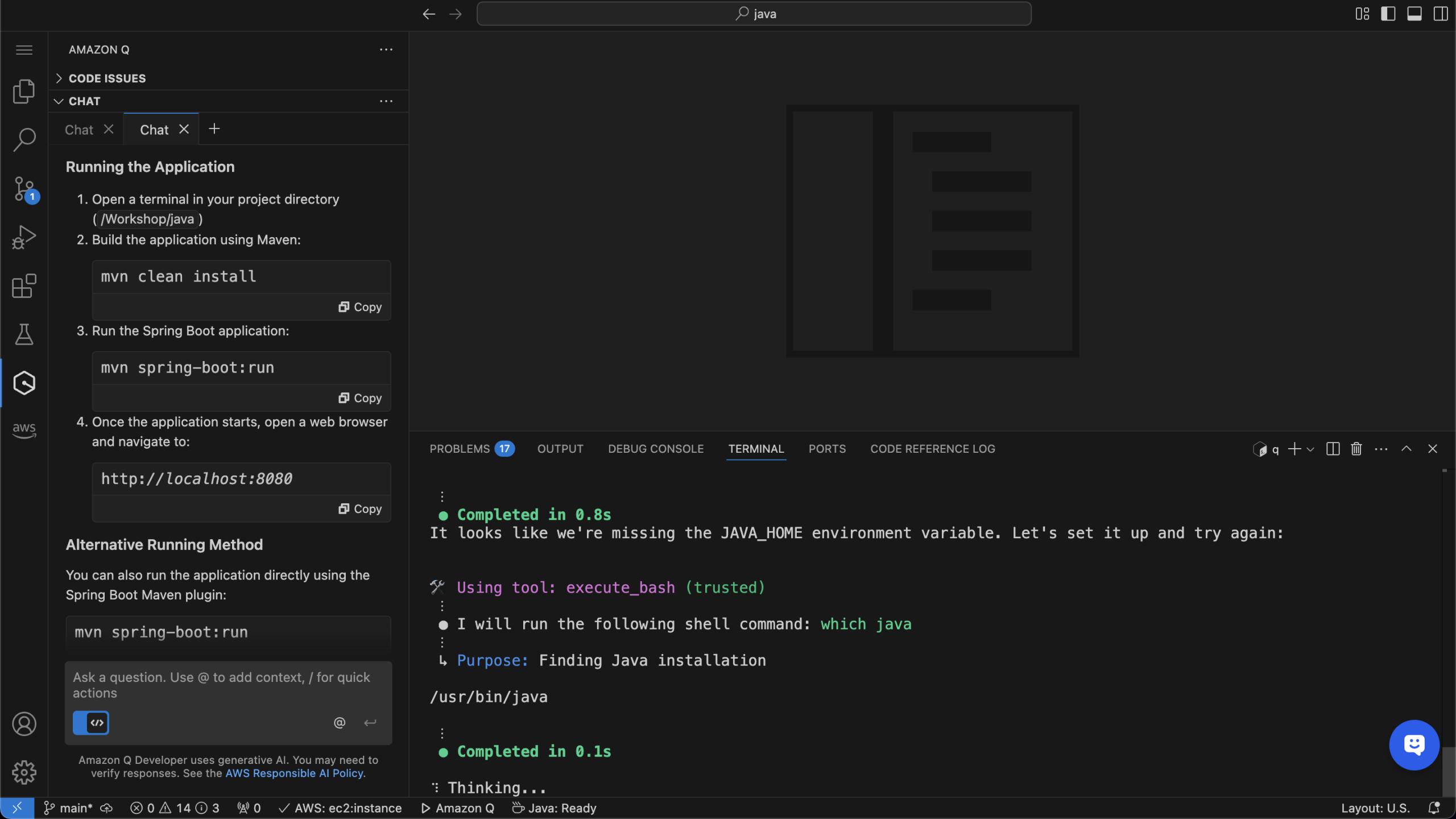
Wordle-like game where players have 5 attempts to guess a 6-letter word, with feedback provided after each guess.

Let's run the application. Since I don't see a pre-built JAR file in the target directory, we'll need to build and run it using Maven:

✖ Using tool: execute_bash
⋮
● I will run the following shell command:
`cd /Workshop/java && mvn spring-boot:run`
⋮
↳ Purpose: Building and running the Spring Boot application using Maven

Allow this action? Use 't' to trust (always allow) this tool for the session. [y/n/t]:

> t 



AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Chat X Chat X +

Running the Application

1. Open a terminal in your project directory
(/Workshop/java)
2. Build the application using Maven:

```
mvn clean install
```


3. Run the Spring Boot application:

```
mvn spring-boot:run
```


4. Once the application starts, open a web browser and navigate to:
<http://localhost:8080>

Alternative Running Method

You can also run the application directly using the Spring Boot Maven plugin:

```
mvn spring-boot:run
```

Ask a question. Use @ to add context, / for quick actions

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS 1 CODE REFERENCE LOG

index: 30
Total words: 37
Words loaded from classpath resource: words.txt
2025-05-31 17:03:47.639 INFO 22963 --- [
index: 18
2025-05-31 17:03:47.986 INFO 22963 --- [
 endpoint(s) beneath base path '/actuator'
2025-05-31 17:03:48.007 INFO 22963 --- [
 started on port(s): 8090 (http) with context path ''
2025-05-31 17:03:48.017 INFO 22963 --- [
 refreshed
2025-05-31 17:03:48.026 INFO 22963 --- [
 custom documentation plugin(s)
2025-05-31 17:03:48.030 INFO 22963 --- [
 or api listing references
2025-05-31 17:03:48.036 INFO 22963 --- [
 ordsApplication in 1.487 seconds (JVM running for 1.701)

main] com.sample.qwords.repository.WordList : Selected i
main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 1
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat sta
main] d.s.w.p.DocumentationPluginsBootstrapper : Context re
main] d.s.w.p.DocumentationPluginsBootstrapper : Found 101
main] s.d. ⓘ Your application running on port 8090 is available. See all ↗
forwarded ports
main] com.





Words

Please enter your name:

Play



Words

Please enter your name:

Play



Words

Make your first guess!

Enter a 6-letter word.

MAKE GUESS

Attempts: 0



Words

Make your first guess!

animal

MAKE GUESS

Attempts: 0



Congratulations, you guessed the word!

PLAY AGAIN

Correct Word: animal

Result:



Attempts: 1

AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Chat X Chat X +

Running the Application

1. Open a terminal in your project directory
(/Workshop/java)
2. Build the application using Maven:

```
mvn clean install
```

Copy

3. Run the Spring Boot application:

```
mvn spring-boot:run
```

Copy

4. Once the application starts, open a web browser and navigate to:

```
http://localhost:8080
```

Copy

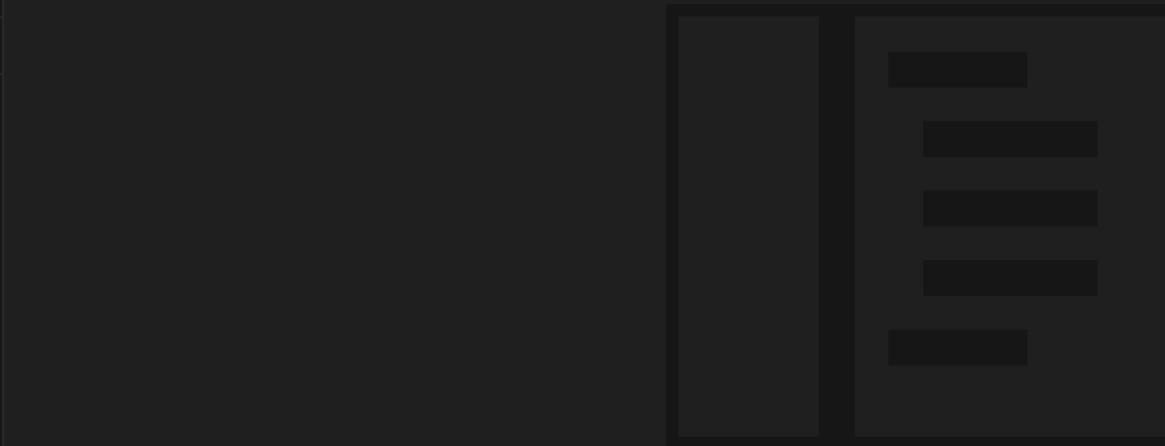
Alternative Running Method

You can also run the application directly using the Spring Boot Maven plugin:

```
mvn spring-boot:run
```

Ask a question. Use @ to add context, / for quick actions

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

Word: animal

Guess: animal

Evaluating current char from guess to word: a->a
 Evaluating current char from guess to word: n->n
 Evaluating current char from guess to word: i->i
 Evaluating current char from guess to word: m->m
 Evaluating current char from guess to word: a->a
 Evaluating current char from guess to word: l->l

2025-05-31 17:04:27.239 INFO 22963 --- [nio-8090-exec-9] c.s.qwords.controller.GameController : User Michael made a guess: animal

2025-05-31 17:04:27.239 INFO 22963 --- [nio-8090-exec-9] c.s.qwords.controller.GameController : Guess: animal, model: {word=animal, attempts=0}, user: Michael ^C

>



Fix a Bug



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

(Optional) Additional Languages

▼ Code Comprehension and Summarization

Task 1: Learn about Amazon Q

Task 2: Learn About the Application Structure

Task 3: Create documentation

▼ Code Debugging, Testing, and Optimization

Task 1: Run and Play Q-Words

Task 2: Identify and Fix a Bug with Amazon Q

Task 3: Writing Unit Tests with Amazon Q Developer

Task 4: Review your code with Amazon Q Developer

► Natural Language to New Feature

► Application Modernization and Transformation

► Generative AI for Accelerating Software Development

Resource Cleanup

Wrapping Up: Next Steps and Feedback

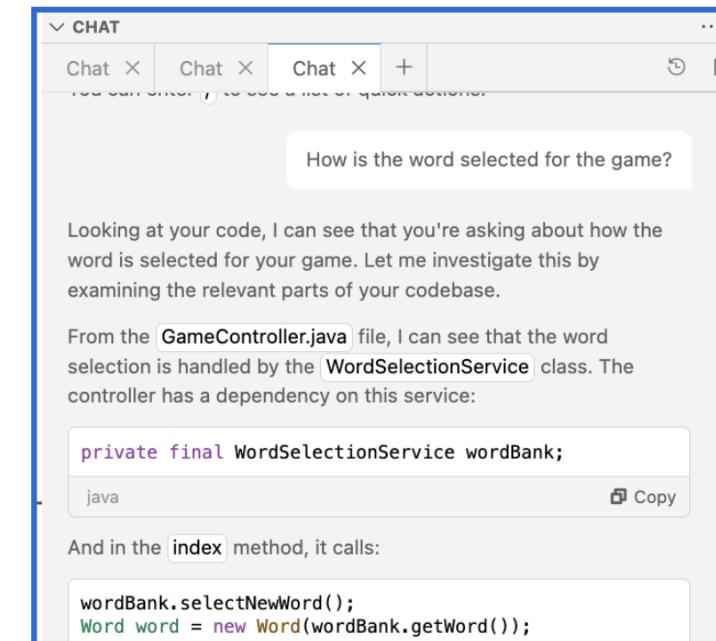
► AWS account access

► Content preferences

Exit event

Activity 1: Investigate the Word Selection Logic

1. Activate the `GameController.java` tab (if you had closed the tab, reopen it from the `src/main/java/com/sample/qwords/controller` directory).
2. In the Amazon Q Developer chat input box, add a prompt to discover how the word is selected for the game: How is the word selected for the game?



Note: Amazon Q chat tries to be efficient in gathering context only when needed. If you see it did not analyze the necessary files, just use `@workspace` or explicitly ask it to analyze the project.

Tip: If you are willing to wait a bit longer on every prompt (or in specific situations), create a rule under `.amazonq/rules/project-context.md` where you instruct Amazon Q to analyze the project contents when necessary to response to a prompt.



The screenshot shows a Java development environment with the following details:

- EXPLORER** panel on the left:

 - Java project structure:
 - .amazonq/rules
 - devfile-guidance.md
 - project-context.md
 - .vscode
 - docs
 - prompts
 - src
 - main
 - java/com/sample/qwords
 - controller
 - Selected file: GameController.java (highlighted with a red arrow)
 - GlobalExceptionHandler.java
 - HomeController.java
 - model
 - GameStatus.java
 - Word.java
 - repository
 - service
 - utils
 - QWordsApplication.java
 - resources
 - test
 - target
 - .gitignore
 - pom.xml
 - README.md
 - README.md.example

- GameController.java** file in the center panel:

```
public class GameController {
    public String index(@RequestParam String user, Model model) {
        int attempts = getAttempts(model);
        return "game";
    }
    /**
     * Process a guess attempt for the game
     */
    @PostMapping("/game")
    public String makeGuess(String guess, Model model, String user) {
        Word word = new Word((String)model.getAttribute("word"));
        String selectedWord = word.getWord();
        int attempts = getAttempts(model);
        String result = word.getInfo(guess);

        log.info("User {} made a guess: {}", user, guess);
        log.info("Guess: {}, model: {}, user: {}", guess, model, user);

        attempts = addAttempt(attempts);

        model.addAttribute("user", user);
        model.addAttribute("guess", guess);
        model.addAttribute("result", result);
        model.addAttribute("attempts", attempts);

        if (guess.equalsIgnoreCase(selectedWord)) {
            model.addAttribute("status", GameStatus.SUCCESS);
            model.addAttribute("message", "Congratulations, you guessed the word!");
            return "game";
        }
    }
}
```

- Bottom status bar:** main* AWS: ec2:instance Java: Ready Workshop Participant (3 days ago) Ln 51, Col 1 Spaces: 4 UTF-8 LF {} Java Layout: U.S.



▼ Running the Workshop in your own AWS Account

[Configure your IDE](#)[Amazon Q Authentication Setup](#)[\(Optional\) Additional Languages](#)

▼ Code Comprehension and Summarization

[Task 1: Learn about Amazon Q](#)[Task 2: Learn About the Application Structure](#)[Task 3: Create documentation](#)

▼ Code Debugging, Testing, and Optimization

[Task 1: Run and Play Q-Words](#)[Task 2: Identify and Fix a Bug with Amazon Q](#)[Task 3: Writing Unit Tests with Amazon Q Developer](#)

▼ AWS account access

[Open AWS console
\(us-east-1\)](#)[Get AWS CLI credentials](#)

▼ Content preferences

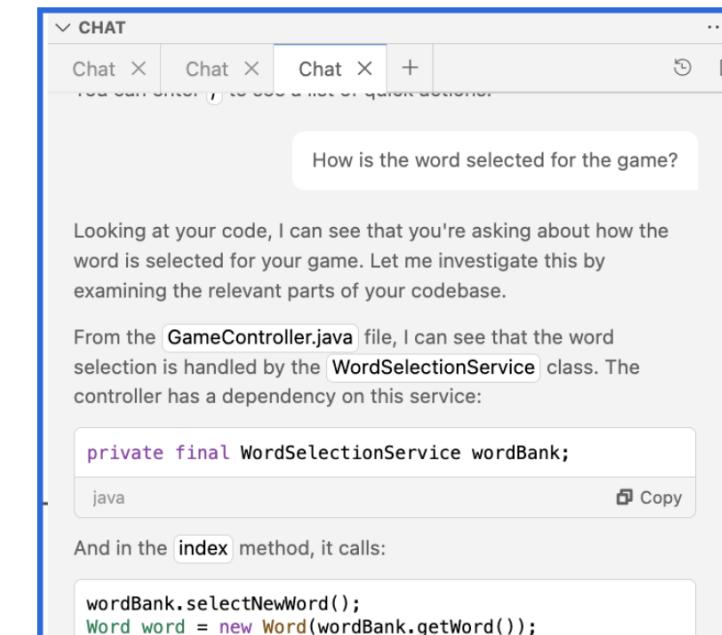
Language

English

Exit event

Activity 1: Investigate the Word Selection Logic

1. Activate the `GameController.java` tab (if you had closed the tab, reopen it from the `src/main/java/com/sample/qwords/controller` directory)
2. In the developer chat input box, add a prompt to discover how the word is selected for the game: How is the word selected for the game? 



The screenshot shows a developer chat interface with three tabs labeled "Chat". A message box contains the prompt: "How is the word selected for the game?". The AI response below states: "Looking at your code, I can see that you're asking about how the word is selected for your game. Let me investigate this by examining the relevant parts of your codebase." It continues: "From the `GameController.java` file, I can see that the word selection is handled by the `WordSelectionService` class. The controller has a dependency on this service:" followed by the Java code:

```
private final WordSelectionService wordBank;
```

 and "And in the `index` method, it calls:" followed by the code:

```
wordBank.selectNewWord();  
Word word = new Word(wordBank.getWord());
```

Note: Amazon Q chat tries to be efficient in gathering context only when needed. If you see it did not analyze the necessary files, just use `@workspace` or explicitly ask it to analyze the project.

Tip: If you are willing to wait a bit longer on every prompt (or in specific situations), create a rule under `.amazonq/rules/project-context.md` where you instruct Amazon Q to analyze the project contents when necessary to response to a prompt



The screenshot shows a developer environment with a Java code editor and a chat interface for Amazon Q.

Code Editor:

- File: GameController.java
- Path: src > main > java > com > sample > qwords > controller > GameController.java
- Content:

```
1 package com.sample.qwords.controller; Amazon Q Tip 1/3: Start typing to get suggestions
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.Model;
5 import org.springframework.web.bind.annotation.GetMapping;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RequestParam;
8 import org.springframework.web.bind.annotation.SessionAttributes;
9
10 import com.sample.qwords.model.GameStatus;
11 import com.sample.qwords.model.Word;
12 import com.sample.qwords.service.WordSelectionService;
13
14 import io.micrometer.core.lang.Nullable;
15 import lombok.extern.log4j.Log4j2;
16
17 /**
18 * A Spring MVC controller class that handles the game logic and user
19 * interactions.
20 */
21 @Controller
22 @Log4j2
23 @SessionAttributes({"word", "attempts"})
24 public class GameController {
25
26     private final WordSelectionService wordBank;
27
28     public GameController(WordSelectionService wordBank) {
29         this.wordBank = wordBank;
30     }
31 }
```

Amazon Q Chat:

- Message: "How is the word selected for the game?" (highlighted with a red arrow)
- Response: "The word is selected by the WordSelectionService, which implements the WordSelectionService interface. This service provides methods for selecting words based on various criteria such as difficulty level, popularity, or random selection. It interacts with a database or external API to fetch the word and store it in session attributes for the user's attempts."
- Buttons: A blue message icon, an '@' icon, and a left arrow icon.
- Footer: "Amazon Q Developer uses generative AI. You may need to verify responses. See the AWS Responsible AI Policy."

IDE UI Elements:

- Sidebar icons: File, Code Issues, Chat, Search, Project, AWS.
- Top bar: Back/Forward, Search (java), and system status.
- Bottom bar: Main tab (main*), AWS instance status, Java status, and file statistics.

The screenshot shows a dark-themed IDE interface with several panels and toolbars.

Left Sidebar:

- Amazon Q icon
- > CODE ISSUES
- CHAT (selected tab)
- Chat X
- Chat X
- Chat X
- +
- How is the word selected for the game?
- > Context
- I'll help you understand how the word is selected for the Q-Words game. Let me examine the relevant code to provide a complete answer.
- Looking at the `GameController.java` file, I can see that the word selection happens through a service called `WordSelectionService`. Let me find and examine this service to understand how words are selected.
- > 1 directory searched
- > 3 files read
- Based on my analysis of the code, here's how the word selection process works in the Q-Words game:
↓
- Ask a question. Use @ to add context, / for quick actions
- Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

Top Bar:

- Back/Forward buttons
- Search bar: java
- Icons for recent files, settings, and help

Code Editor:

File Path: src > main > java > com > sample > qwords > controller > GameController.java > {} com.sample.qwords.controller

Code:

```
1 package com.sample.qwords.controller; Amazon Q Tip 1/3: Start typing to get suggestions
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.Model;
5 import org.springframework.web.bind.annotation.GetMapping;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RequestParam;
8 import org.springframework.web.bind.annotation.SessionAttributes;
9
10 import com.sample.qwords.model.GameStatus;
11 import com.sample.qwords.model.Word;
12 import com.sample.qwords.service.WordSelectionService;
13
14 import io.micrometer.core.lang.Nullable;
15 import lombok.extern.log4j.Log4j2;
16
17 /**
18 * A Spring MVC controller class that handles the game logic and user
19 * interactions.
20 */
21 @Controller
22 @Log4j2
23 @SessionAttributes({"word", "attempts"})
24 public class GameController {
25
26     private final WordSelectionService wordBank;
27
28     public GameController(WordSelectionService wordBank) {
29         this.wordBank = wordBank;
30     }
31 }
```

Bottom Status Bar:

- main* ↻
- ✖ 0 ⚠ 14 ⓘ 3 ⚡ 0 ✓ AWS: ec2:instance
- ▶ Amazon Q
- ⌚ Java: Ready
- Ln 1, Col 1 Spaces: 4 UTF-8 LF {} Java Layout: U.S.

The screenshot shows a dark-themed IDE interface. On the left, there's a sidebar with various icons: a gear, a person, a search bar, a file, a list, and others. Below the sidebar, the text "AMAZON Q" is displayed. The main area has tabs for "CODE ISSUES" and "CHAT". Under "CHAT", there are three tabs: "Chat X", "Chat X", and "Chat X". A plus sign icon is also present.

In the center, the code editor displays `GameController.java`. The code is a Spring MVC controller for a game logic. It imports several Spring framework annotations and classes, including `Controller`, `Model`, `GetMapping`, `PostMapping`, `RequestParam`, `SessionAttributes`, `GameStatus`, `Word`, `WordSelectionService`, `Nullable`, and `Log4j2`. The class is annotated with `@Controller`, `@Log4j2`, and `@SessionAttributes`. The constructor takes a `WordSelectionService` dependency and initializes it.

A tooltip from "Amazon Q" appears over the code, reading: "Tip 1/3: Start typing to get suggestions".

The bottom of the screen shows the status bar with the file name "main*", line count "0", column count "14", character count "3", and the message "AWS: ec2:instance". It also indicates "Java: Ready". The status bar also shows "Ln 1, Col 1", "Spaces: 4", "UTF-8", "LF", "{} Java", and "Layout: U.S.". A small "AWS" logo is also visible in the status bar.

On the right side of the screen, there's a vertical panel showing a detailed analysis or history of the code, with multiple entries listed vertically.

The screenshot shows a dark-themed IDE interface with a sidebar on the left containing various icons for AWS services like Lambda, CloudWatch, and CloudFront. The main area has a search bar at the top with the text 'java'. A tab labeled 'GameController.java' is active. The code editor displays the following Java code:

```
src > main > java > com > sample > qwords > controller > GameController.java > {} com.sample.qwords.controller
1 package com.sample.qwords.controller; Amazon Q Tip 1/3: Start typing to get suggestions
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.Model;
5 import org.springframework.web.bind.annotation.GetMapping;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RequestParam;
8 import org.springframework.web.bind.annotation.SessionAttributes;
9
10 import com.sample.qwords.model.GameStatus;
11 import com.sample.qwords.model.Word;
12 import com.sample.qwords.service.WordSelectionService;
13
14 import io.micrometer.core.lang.Nullable;
15 import lombok.extern.log4j.Log4j2;
16
17 /**
18 * A Spring MVC controller class that handles the game logic and user
19 * interactions.
20 */
21 @Controller
22 @Log4j2
23 @SessionAttributes({"word", "attempts"})
24 public class GameController {
25
26     private final WordSelectionService wordBank;
27
28     public GameController(WordSelectionService wordBank) {
29         this.wordBank = wordBank;
30     }
31 }
```

The code uses annotations from the Spring Framework and Lombok. It defines a controller class named 'GameController' with methods for handling GET and POST requests. The class also uses session attributes for tracking game status and words.

The screenshot shows a dark-themed IDE interface with the following components:

- Sidebar:** On the left, there are several icons: a list icon (AMAZON Q), a file icon (CODE ISSUES), a person icon (CHAT), a search icon, a gear icon (aws), and a user icon.
- Search Bar:** At the top center is a search bar with the text "java".
- Code Editor:** The main area displays the code for `GameController.java`.

```
1 package com.sample.qwords.controller; Amazon Q Tip 1/3: Start typing to get suggestions
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.Model;
5 import org.springframework.web.bind.annotation.GetMapping;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RequestParam;
8 import org.springframework.web.bind.annotation.SessionAttributes;
9
10 import com.sample.qwords.model.GameStatus;
11 import com.sample.qwords.model.Word;
12 import com.sample.qwords.service.WordSelectionService;
13
14 import io.micrometer.core.lang.Nullable;
15 import lombok.extern.log4j.Log4j2;
16
17 /**
18 * A Spring MVC controller class that handles the game logic and user
19 * interactions.
20 */
21 @Controller
22 @Log4j2
23 @SessionAttributes({"word", "attempts"})
24 public class GameController {
25
26     private final WordSelectionService wordBank;
27
28     public GameController(WordSelectionService wordBank) {
29         this.wordBank = wordBank;
30     }
31 }
```
- Code Snippet Overlay:** A floating panel on the left contains the following code:

```
currentWord = this.wordList.get(ind)
java
```

A "Copy" button is visible next to the Java button.
- Feedback Panel:** A small box at the bottom left says "Ask a question. Use @ to add context, / for quick actions". It includes a blue "Ask" button with a person icon, a text input field starting with "@", and a backspace key icon.
- Footer:** The bottom navigation bar includes tabs for "main*", "AWS: ec2:instance", "Amazon Q", "Java: Ready", and status indicators for "Ln 1, Col 1", "Spaces: 4", "UTF-8", "LF", "{} Java", and "Layout: U.S."

The screenshot shows the AWS Toolkit for VS Code interface. The Explorer sidebar on the left displays the project structure, including Java files like GameController.java, WordSelectionService.java, and QWordsApplication.java. A red arrow points to the WordSelectionService.java file in the list. The main editor area shows the code for WordSelectionService.java, which handles the selection of a random word from a word list. The status bar at the bottom indicates "Java: Ready".

```
package com.sample.qwords.service;
import org.springframework.stereotype.Service;
import com.sample.qwords.repository.WordList;

/**
 * A service class that handles the selection of a random word from a word list.
 */
@Service
public class WordSelectionService {

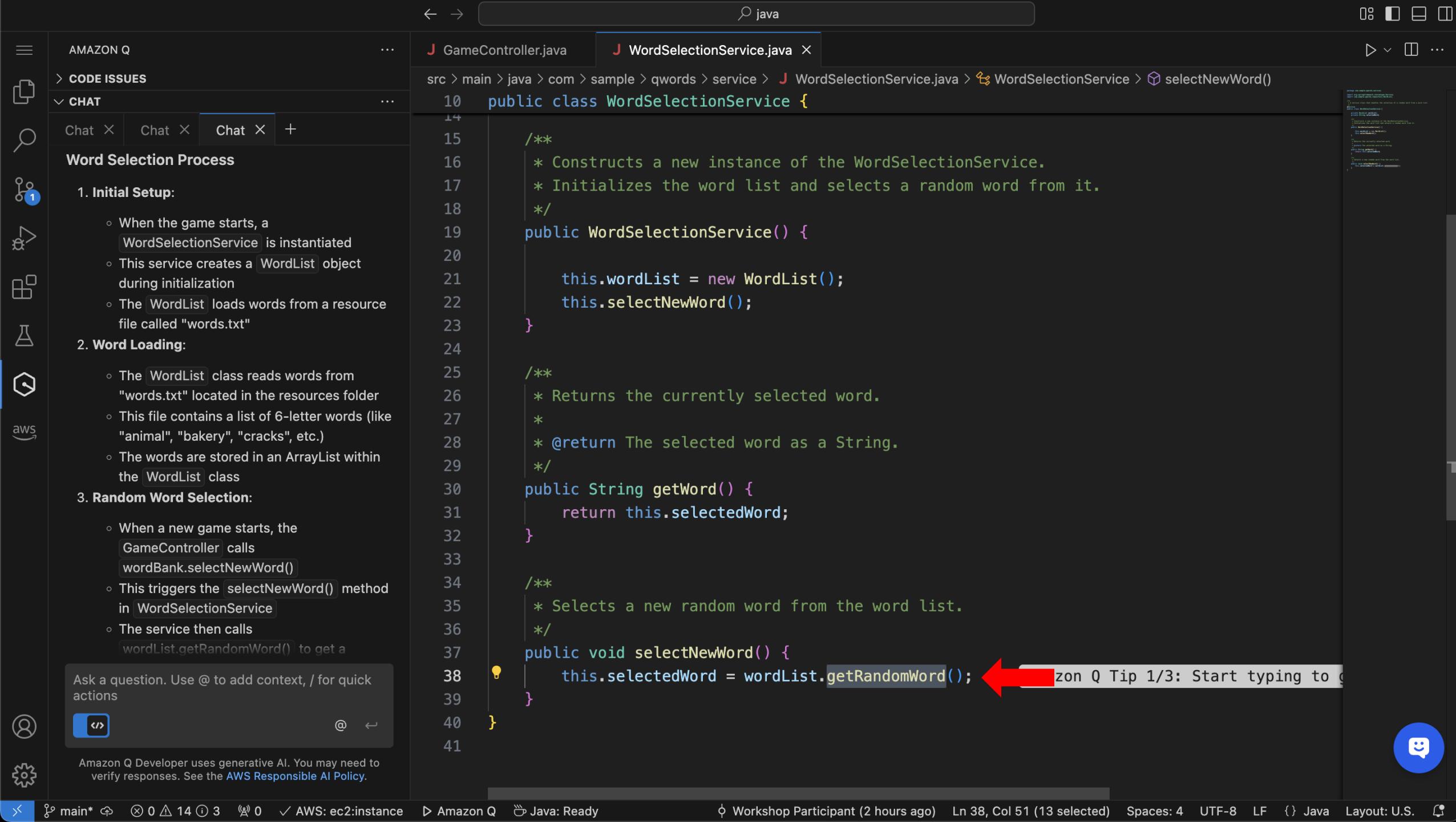
    private WordList wordList;
    private String selectedWord;

    /**
     * Constructs a new instance of the WordSelectionService.
     * Initializes the word list and selects a random word from it.
     */
    public WordSelectionService() {

        this.wordList = new WordList();
        this.selectNewWord();
    }

    /**
     * Returns the currently selected word.
     *
     * @return The selected word as a String.
     */
    public String getWord() {

```



AMAZON Q

> CODE ISSUES

CHAT

Chat X Chat X Chat X +

Word Selection Process

- Initial Setup:**
 - When the game starts, a `WordSelectionService` is instantiated
 - This service creates a `WordList` object during initialization
 - The `WordList` loads words from a resource file called "words.txt"
- Word Loading:**
 - The `WordList` class reads words from "words.txt" located in the resources folder
 - This file contains a list of 6-letter words (like "animal", "bakery", "cracks", etc.)
 - The words are stored in an `ArrayList` within the `WordList` class
- Random Word Selection:**
 - When a new game starts, the `GameController` calls `wordBank.selectNewWord()`
 - This triggers the `selectNewWord()` method in `WordSelectionService`
 - The service then calls `wordList.getRandomWord()` to get a

Ask a question. Use @ to add context, / for quick actions

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

java

GameController.java WordSelectionService.java WordList.java

src > main > java > com > sample > qwords > repository > WordList.java > WordList > getRandomWord()

```

20  public class WordList {
144     * Shuffles the word list.
145     */
146     public void shuffleWordList() {
147         Collections.shuffle(wordList);
148     }
149
150     public List<String> getWordsByLength(int length) {
151         return wordList.stream().filter(word -> word.length() == length).collect(Collectors.to
152     }
153
154     public String getRandomWord() { Amazon Q Tip 1/3: Start typing to get suggestions ([E]
155         Random random = new Random();
156         int index = random.nextInt(wordList.size());
157         // log the selected index
158         logger.info("Selected index: " + index);
159
160         // track the current and last word returned using
161         // lastWord and currentWord class properties
162         if (wordList.isEmpty()) {
163             return null;
164         }
165
166         // if the currentWord is not null, store it in the lastWord variable before
167         // getting a new word.
168         if (currentWord != null) {
169             lastWord = currentWord;
170         }
171         currentWord = this.wordList.get(index:0);
172         return currentWord;
173
174     }

```

AMAZON Q

> CODE ISSUES

CHAT

Chat X Chat X Chat X +

5. Word Tracking:

- The `WordList` class keeps track of the current and previous words using `currentWord` and `lastWord` variables
- When a new word is selected, the previous word is stored in `lastWord`

Summary

The word selection process is intended to be random, but there's a bug in the implementation. Instead of using the randomly generated index to select a word from the list, the code always returns the first word in the list. This means all players would get the same word ("animal" based on the current `words.txt` file) rather than a random selection.

To fix this issue, the `getRandomWord()` method in `WordList.java` should be modified to use the random index it generates:

```
currentWord = this.wordList.get(index);  
java ⚡ Copy
```

Ask a question. Use @ to add context, / for quick actions



@ ←

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

```
J GameController.java J WordSelectionService.java J WordList.java X  
src > main > java > com > sample > qwords > repository > J WordList.java > WordList > getRandomWord()  
20 public class WordList {  
152     }  
153  
154     public String getRandomWord() {  
155         Random random = new Random();  
156         int index = random.nextInt(wordList.size());  
157         // log the selected index  
158         logger.info("Selected index: " + index);  
159  
160         // track the current and last word returned using  
161         // lastWord and currentWord class properties  
162         if (wordList.isEmpty()) {  
163             return null;  
164         }  
165  
166         // if the currentWord is not null, store it in the lastWord variable before  
167         // getting a new word.  
168         if (currentWord != null) {  
169             lastWord = currentWord;  
170         }  
171         currentWord = this.wordList.get(index);  
172         return currentWord;  
173     }  
174  
175     public List<String> getWords() {  
176         return new ArrayList<>(wordList);  
177     }  
178  
179     public static void resetInstance() {  
180         if (instance != null) {  
181             instance = null;
```



Amazon Q Tip 1/3: Start typing to get s

▼ Running the Workshop in your own AWS Account

[Configure your IDE](#)[Amazon Q Authentication Setup](#)[\(Optional\) Additional Languages](#)

▼ Code Comprehension and Summarization

[Task 1: Learn about Amazon Q](#)[Task 2: Learn About the Application Structure](#)[Task 3: Create documentation](#)

▼ Code Debugging, Testing, and Optimization

[Task 1: Run and Play Q-Words](#)[Task 2: Identify and Fix a Bug with Amazon Q](#)[Task 3: Writing Unit Tests with Amazon Q Developer](#)

▼ AWS account access

[Open AWS console
\(us-east-1\)](#)[Get AWS CLI credentials](#)

▼ Content preferences

Language

English

Exit event

Activity 3: Fix the Bug with Amazon Q



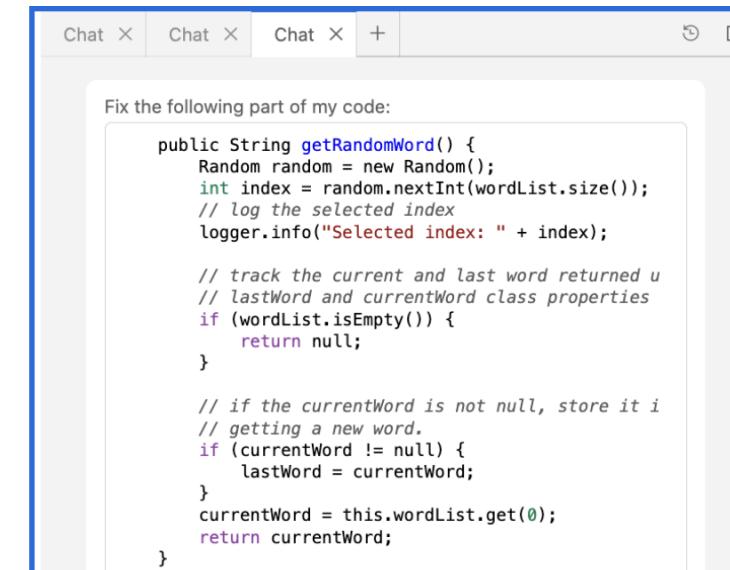
The response to the previous prompts may already contain a suggested fix, or it may have fixed it for you if you were in agentic coding mode. If not, let's try other methods to fix this bug.

Fix Context Menu

1. Open `WordList.java` class file in the repository directory and find the `getRandomWord` function.
2. You should see that it only returns the first item from the list defined in the constructor.
3. Highlight the problem function and right-click **Amazon Q → Fix** or **Amazon Q → Send to Prompt**. 

Note

If you need to provide more context or example errors, use **Send to Prompt** and type something similar to "This function is producing the following error: Error:.... How can I fix it? .



Fix the following part of my code:

```
public String getRandomWord() {
    Random random = new Random();
    int index = random.nextInt(wordList.size());
    // log the selected index
    logger.info("Selected index: " + index);

    // track the current and last word returned
    // lastWord and currentWord class properties
    if (wordList.isEmpty()) {
        return null;
    }

    // if the currentWord is not null, store it in
    // getting a new word.
    if (currentWord != null) {
        lastWord = currentWord;
    }
    currentWord = this.wordList.get(0);
    return currentWord;
}
```



The screenshot shows a developer's workspace with an IDE interface. On the left, there's a sidebar with various icons and sections like 'AMAZON Q', 'CODE ISSUES', and 'CHAT'. The main area displays Java code for 'WordList.java'.

Code View:

```
src > main > java > com > sample > qwords > repository > WordList.java > WordList
20 public class WordList {
152     ...
153
154     public String getRandomWord() { Amazon Q Tip 1/3: Start typing to get suggestions ([]
155         Random random = new Random();
156         int index = random.nextInt(wordList.size());
157         // log the selected index
158         logger.info("Selected index: " + index);
159
160         // track the current and last word returned using
161         // lastWord and currentWord class properties
162         if (wordList.isEmpty()) {
163             return null;
164         }
165
166         // if the currentWord is not null, store it in the lastWord variable before
167         // getting a new word.
168         if (currentWord != null) {
169             lastWord = currentWord;
170         }
171         currentWord = this.wordList.get(index:0);
172         return currentWord;
173     }
174
175     public List<String> getWords() {
176         return new ArrayList<>(wordList);
177     }
178
179     public static void resetInstance() {
180         if (instance != null) {
181             instance = null;
182         }
183     }
184 }
```

Annotations:

- Line 154: 'Amazon Q Tip 1/3: Start typing to get suggestions ([])'
- Line 166: 'if the currentWord is not null, store it in the lastWord variable before getting a new word.'

Amazon Q Sidebar:

- Section: 5. Word Tracking:
 - The `WordList` class keeps track of the current and previous words using `currentWord` and `lastWord` variables
 - When a new word is selected, the previous word is stored in `lastWord`
- Summary:

The word selection process is intended to be random, but there's a bug in the implementation. Instead of using the randomly generated index to select a word from the list, the code always returns the first word in the list. This means all players would get the same word ("animal" based on the current words.txt file) rather than a random selection.

To fix this issue, the `getRandomWord()` method in `WordList.java` should be modified to use the random index it generates:

```
currentWord = this.wordList.get(index:0);
```

Ask a question. Use @ to add context, / for quick actions

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

AMAZON Q ...

> CODE ISSUES

CHAT Chat X Chat X Chat X +

word in the list

5. Word Tracking:

- The `WordList` class keeps track of the current and previous words using `currentWord` and `lastWord` variables
- When a new word is selected, the previous word is stored in `lastWord`

Summary

The word selection process is intended to be random, but there's a bug in the implementation. Instead of using the randomly generated index to select a word from the list, the code always returns the first word in the list. This means all players would get the same word ("animal" based on the current `words.txt` file) rather than a random selection.

To fix this issue, the `getRandomWord()` method in `WordList.java` should be modified to use the random index it generates:

```
currentWord = this.wordList.get(index)
```

java Copy

Ask a question. Use @ to add context, / for quick actions

@ ↵

Amazon Q Developer uses generative AI. You may need to verify responses. See the AWS Responsible AI Policy.

src > main > java > com > sample > qwords > repository > J

20 public class WordList {
152 }
153
154 public String getRandomWord() {
155 Random random = new Random();
156 int index = random.nextInt();
157 // log the selected index
158 logger.info("Selected index: " + index);
159
160 // track the current and last word
161 // lastWord and currentWord
162 if (wordList.isEmpty()) {
163 return null;
164 }
165
166 // if the currentWord is no longer
167 // getting a new word.
168 if (currentWord != null) {
169 lastWord = currentWord;
170 }
171 currentWord = this.wordList.get(index);
172 return currentWord;
173 }
174
175 public List<String> getWords()
176 return new ArrayList<>(wordList);
177 }
178
179 public static void resetInstance()
180 if (instance != null) {
181 instance = null;

J GameController.java J WordSelectionService.java J WordList.java X

Go to Definition ⌘F12
Go to Declaration
Go to Type Definition
Go to Implementations ⌘F12
Go to References ⌘F12
Go to Super Implementation
Go to Test
Peek >
Find All References ⌘⌥F12
Find All Implementations
Show Call Hierarchy ⌘⌥H
Show Type Hierarchy
Rename Symbol F2
Change All Occurrences ⌘F2
Format Document ⌘⌥F
Format Document With...
Format Selection ⌘K ⌘F
Refactor... ⌘⌃R
Source Action...
Cut
Copy
Paste
Amazon Q >
Run Java
Debug Java
Command Palette... ⌘⌘P

ping to get suggestions ([B] variable before [variable before])

Explain ⌘⌘E
Refactor ⌘⌘U
Fix ⌘⌘Y ← Red arrow
Optimize ⌘⌘A
Generate Tests ⌘⌘T
Send to prompt ⌘⌘Q
Inline chat ⌘I

Spaces: 4 UTF-8 LF {} Java Layout: U.S. ↵

AMAZON Q

> CODE ISSUES

✓ CHAT

Chat X Chat X Chat X +

Fix the following part of my code:

```
public String getRandomWord() {
    Random random = new Random();
    int index = random.nextInt(wordList.size());
    // log the selected index
    logger.info("Selected index: " + index);

    // track the current and last word
    if (wordList.isEmpty())
        return null;

    // if the currentWord is null, getting a new word.
    if (currentWord != null)
        lastWord = currentWord;
    currentWord = this.wordList.get(index);
    return currentWord;
}
```

> Context

I'll fix the `getRandomWord()` method. The issue is that it's always returning the first word in the list (index 0) instead of using the randomly generated

Ask a question. Use @ to add context, / for quick actions

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

java

GameController.java WordSelectionService.java WordList.java

src > main > java > com > sample > qwords > repository > WordList.java > WordList

20 public class WordList {
152 }
153
154 public String getRandomWord() { Amazon Q Tip 1/3: Start typing to get suggestions ([
155 Random random = new Random();
156 int index = random.nextInt(wordList.size());
157 // log the selected index
158 logger.info("Selected index: " + index);
159
160 // track the current and last word returned using
161 // lastWord and currentWord class properties
162 if (wordList.isEmpty()) {
163 return null;
164 }
165
166 // if the currentWord is not null, store it in the lastWord variable before
167 // getting a new word.
168 if (currentWord != null) {
169 lastWord = currentWord;
170 }
171 currentWord = this.wordList.get(index);
172 return currentWord;
173 }
174
175 public List<String> getWords() {
176 return new ArrayList<>(wordList);
177 }
178
179 public static void resetInstance() {
180 if (instance != null) {
181 instance = null;
182 }
183 }
184 }

Java: Ready

Workshop Participant (2 hours ago)

Ln 154, Col 1 (665 selected)

Spaces: 4

UTF-8

LF

{}

Java

Layout: U.S.

AMAZON Q

> CODE ISSUES

CHAT Chat X Chat X +

I'll fix the `getRandomWord()` method. The issue is that it's always returning the first word in the list (index 0) instead of using the randomly generated index.

Fixing the `getRandomWord` method to use the randomly generated index instead of always returning the first word

WordList.java +5 -10 Undo

I've fixed the bug in the `getRandomWord()` method. Here's what I changed:

1. Fixed the main issue: Changed `currentWord = this.wordList.get(0)` to `currentWord = this.wordList.get(index)` so it actually uses the random index that was generated.
2. Improved the code flow:
 - Moved the empty list check to the beginning of the method to fail fast
 - Simplified the logic by removing unnecessary comments and reordering statements for better readability

Now the game will properly select a random word

Ask a question. Use @ to add context, / for quick actions

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

java

src > main > java > com > sample > qwords > repository > WordList.java > WordList

```
public class WordList {  
    ...  
    public String getRandomWord() {  
        if (wordList.isEmpty()) {  
            return null;  
        }  
        Random random = new Random();  
        int index = random.nextInt(wordList.size());  
        logger.info("Selected index: " + index);  
        if (currentWord != null) {  
            lastWord = currentWord;  
        }  
        currentWord = this.wordList.get(index);  
        return currentWord;  
    }  
  
    public List<String> getWords() {  
        return new ArrayList<>(wordList);  
    }  
  
    public static void resetInstance() {  
        if (instance != null) {  
            instance = null;  
        }  
        throw new UnsupportedOperationException("Unimplemented method 'resetInstance'");  
    }  
  
    public String getLastWord() {  
        ...  
    }  
}
```

Amazon Q Tip 1/3: Start typing to get suggestions ([)

main* ↻ ⑧ 1 ▲ 14 ③ ④ 0 ✓ AWS: ec2:instance ▶ Amazon Q ☁ Java: Ready ⚡ Workshop Participant (2 hours ago) Ln 154, Col 1 (417 selected) Spaces: 4 UTF-8 LF {} Java Layout: U.S. 🔍

AMAZON Q

> CODE ISSUES

CHAT Chat X Chat X Chat X +

Fixing the getRandomWord method to use the randomly generated index instead of always returning the first word

WordList.java +5 -10 ← Undo

I've fixed the bug in the `getRandomWord()` method. Here's what I changed:

1. Fixed the main issue: Changed `currentWord = this.wordList.get(0)` to `currentWord = this.wordList.get(index)` so it actually uses the random index that was generated.
2. Improved the code flow:
 - Moved the empty list check to the beginning of the method to fail fast
 - Simplified the logic by removing unnecessary comments and reordering statements for better readability

Now the game will properly select a random word from the word list each time a new game starts, giving players different words to guess rather than always using the first word in the list.

Ask a question. Use @ to add context, / for quick actions

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

java

src > main > java > com > sample > qwords > repository > WordList.java > WordList

```
public class WordList {  
    ...  
    public String getRandomWord() {  
        if (wordList.isEmpty()) {  
            return null;  
        }  
        Random random = new Random();  
        int index = random.nextInt(wordList.size());  
        logger.info("Selected index: " + index);  
        if (currentWord != null) {  
            lastWord = currentWord;  
        }  
        currentWord = this.wordList.get(index);  
        return currentWord;  
    }  
  
    public List<String> getWords() {  
        return new ArrayList<>(wordList);  
    }  
  
    public static void resetInstance() {  
        if (instance != null) {  
            instance = null;  
        }  
        throw new UnsupportedOperationException("Unimplemented method 'resetInstance'");  
    }  
  
    public String getLastWord() {  
        ...  
    }  
}
```

Amazon Q Tip 1/3: Start typing to get suggestions ([)

main* ↗ ⑧ 1 ▲ 14 ③ ④ 0 ✓ AWS: ec2:instance ▶ Amazon Q ⚡ Java: Ready ⚡ Workshop Participant (2 hours ago) Ln 154, Col 1 (417 selected) Spaces: 4 UTF-8 LF {} Java Layout: U.S.

J GameController.java

J WordSelectionService.java

J WordList.java (Generated by Amazon Q) X

▶ ⌂ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋

```
20 public class WordList {  
  Explorer (⌘E)    public void shuffleWordList() {  
 148 }  
 149  
150     public List<String> getWordsByLength(int length) {  
151         return wordList.stream().filter(word -> word.length() == len  
152     }  
153  
154     public String getRandomWord() {  
155         Random random = new Random();  
156         int index = random.nextInt(wordList.size());  
157         // log the selected index  
158         logger.info("Selected index: " + index);  
159  
160         // track the current and last word returned using  
161         // lastWord and currentWord class properties  
162         if (wordList.isEmpty()) {  
163             return null;  
164         }  
165  
166         // if the currentWord is not null, store it in the lastWord  
167         // getting a new word.  
168  
169         if (currentWord != null) {  
170             lastWord = currentWord;  
171         currentWord = this.wordList.get(0);  
172         return currentWord;  
173     }  
174  
175     public List<String> getWords() {
```

```
20 public class WordList {  
 146     public void shuffleWordList() {  
 148 }  
 149  
150     public List<String> getWordsByLength(int length) {  
151         return wordList.stream().filter(word -> word.length() == len  
152     }  
153  
154     public String getRandomWord() {  
155         if (wordList.isEmpty()) {  
156             return null;  
157         }  
158  
159+         Random random = new Random();  
160+         int index = random.nextInt(wordList.size());  
161+         logger.info("Selected index: " + index);  
162+  
163         if (currentWord != null) {  
164             lastWord = currentWord;  
165         }  
166+         currentWord = this.wordList.get(index);  
167         return currentWord;  
168     }  
169  
170     public List<String> getWords() {
```



A screenshot of a Java code editor interface, likely from a browser-based IDE or developer tool. The top navigation bar shows tabs for "WordList.java" and "QWords". The left sidebar contains a file tree under the "EXPLORER" tab, showing a project structure with Java files like GameController.java, WordSelectionService.java, and WordList.java. The main workspace displays the "WordList.java" code, which includes methods for getting random words and returning a list of words. A code completion tooltip is visible over the "getRandomWord" method.

```
public class WordList {  
    public String getRandomWord() {  
        if (wordList.isEmpty()) {  
            return null;  
        }  
  
        Random random = new Random();  
        int index = random.nextInt(wordList.size());  
        logger.info("Selected index: " + index);  
  
        if (currentWord != null) {  
            lastWord = currentWord;  
        }  
        currentWord = this.wordList.get(index);  
        return currentWord;  
    }  
  
    public List<String> getWords() {  
        return new ArrayList<>(wordList);  
    }  
  
    public static void resetInstance() {  
        if (instance != null) {  
            instance = null;  
        }  
        throw new UnsupportedOperationException(message:"Unimplemented method 'resetInstance'"  
    }  
}
```

The screenshot shows a Java Spring Boot application running in a VS Code environment. The terminal output is as follows:

```
Word: animal
Guess: animal
Evaluating current char from guess to word: a->a
Evaluating current char from guess to word: n->n
Evaluating current char from guess to word: i->i
Evaluating current char from guess to word: m->m
Evaluating current char from guess to word: a->a
Evaluating current char from guess to word: l->l
2025-05-31 17:04:27.239 INFO 22963 --- [nio-8090-exec-9] c.s.qwords.controller.GameController : User Michael
el made a guess: animal
2025-05-31 17:04:27.239 INFO 22963 --- [nio-8090-exec-9] c.s.qwords.controller.GameController : Guess: ani
mal, model: {word=animal, attempts=0}, user: Michael
^C
```

A red arrow points to the last line of the terminal output: "Can you run this Spring Boot application for me?".

EXPLORER

- ...
- JAVA**
 - .amazonq/rules
 - devfile-guidance.md
 - project-context.md
 - > .vscode
 - > prompts
 - src**
 - main**
 - java/com/sample/qwords
 - controller
 - J GameController.java
 - J GlobalExceptionHandler.java
 - J HomeController.java
 - model
 - repository
 - J WordList.java
 - service
 - J WordSelectionService.java
 - utils
 - J QWordsApplication.java
 - > resources
 - > test
 - > target
 - ↳ .gitignore
 - ↳ pom.xml
 - ≡ README.md.example
 - > OUTLINE
 - > TIMELINE
 - > APPLICATION BUILDER
 - > JAVA PROJECTS
 - > MAVEN

PROBLEMS 17 OUTPUT DEBUG CONSOLE **TERMINAL** PORTS 1 CODE REFERENCE LOG

```

index: 21
Total words: 37
Words loaded from classpath resource: words.txt
2025-05-31 17:16:05.932  INFO 28622 --- [
index: 31
2025-05-31 17:16:06.290  INFO 28622 --- [
  endpoint(s) beneath base path '/actuator'
2025-05-31 17:16:06.318  INFO 28622 --- [
  started on port(s): 8090 (http) with context path ''
2025-05-31 17:16:06.324  INFO 28622 --- [
  refreshed
2025-05-31 17:16:06.332  INFO 28622 --- [
  custom documentation plugin(s)
2025-05-31 17:16:06.336  INFO 28622 --- [
  or api listing references
2025-05-31 17:16:06.341  INFO 28622 --- [
  ordsApplication in 1.528 seconds (JVM running for 1.71)

```

main] com.sample.qwords.repository.WordList : Selected i
 main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 1
 main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat sta
 main] d.s.w.p.DocumentationPluginsBootstrapper : Context re
 main] d.s.w.p.DocumentationPluginsBootstrapper : Found 101
 main] s.d. (i) Your application running on port 8090 is available. See all ↗ forwarded ports
 main] com.

Open in Brow





Words

Please enter your name:

Play

Enter your name (1-15 letters)



Words

Please enter your name:

Play



Words

Make your first guess!

Enter a 6-letter word.

MAKE GUESS

Enter a word with 6 letters

Attempts: 0



Try again!

Enter a 6-letter word.

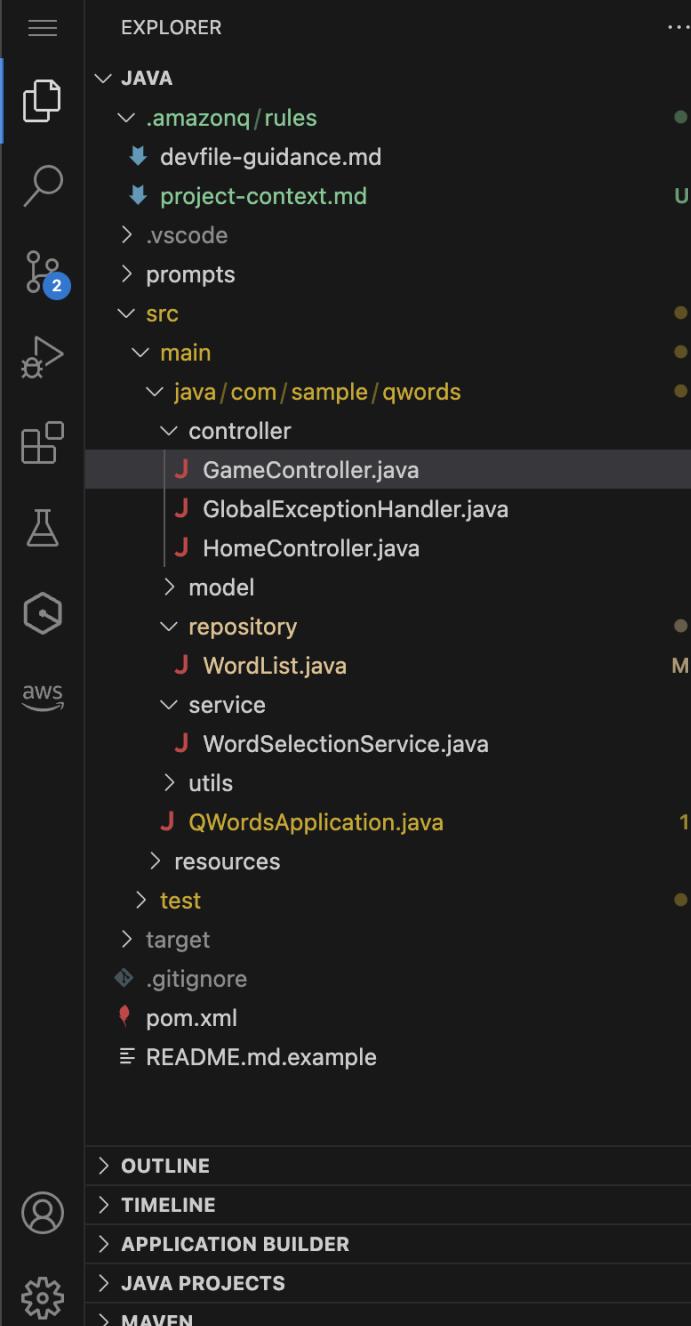
MAKE GUESS

animal

Result:



Attempts: 1



PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

Word: butane

Guess: animal

```
Evaluating current char from guess to word: a->b
Evaluating current char from guess to word: n->u
Evaluating current char from guess to word: i->t
Evaluating current char from guess to word: m->a
Evaluating current char from guess to word: a->n
Evaluating current char from guess to word: l->e
2025-05-31 17:16:58.632 INFO 28622 --- [io-8090-exec-10] c.s.qwords.controller.GameController : User Michael
el made a guess: animal
2025-05-31 17:16:58.632 INFO 28622 --- [io-8090-exec-10] c.s.qwords.controller.GameController : Guess: ani
mal, model: {word=butane, attempts=1}, user: Michael
^C
```

A red arrow points to the '^C' character at the bottom of the terminal output.



Review the Code

(Optional) Additional Languages

▼ Code Comprehension and Summarization

Task 1: Learn about Amazon Q

Task 2: Learn About the Application Structure

Task 3: Create documentation

▼ Code Debugging, Testing, and Optimization

Task 1: Run and Play Q-Words

Task 2: Identify and Fix a Bug with Amazon Q

Task 3: Writing Unit Tests with Amazon Q Developer

Task 4: Review your code with Amazon Q Developer

▼ Natural Language to New Feature

Task 1: Describe the Feature to Amazon Q

Task 2: Crafting Effective Prompts for Amazon Q Developer

Task 3: Migrating Between

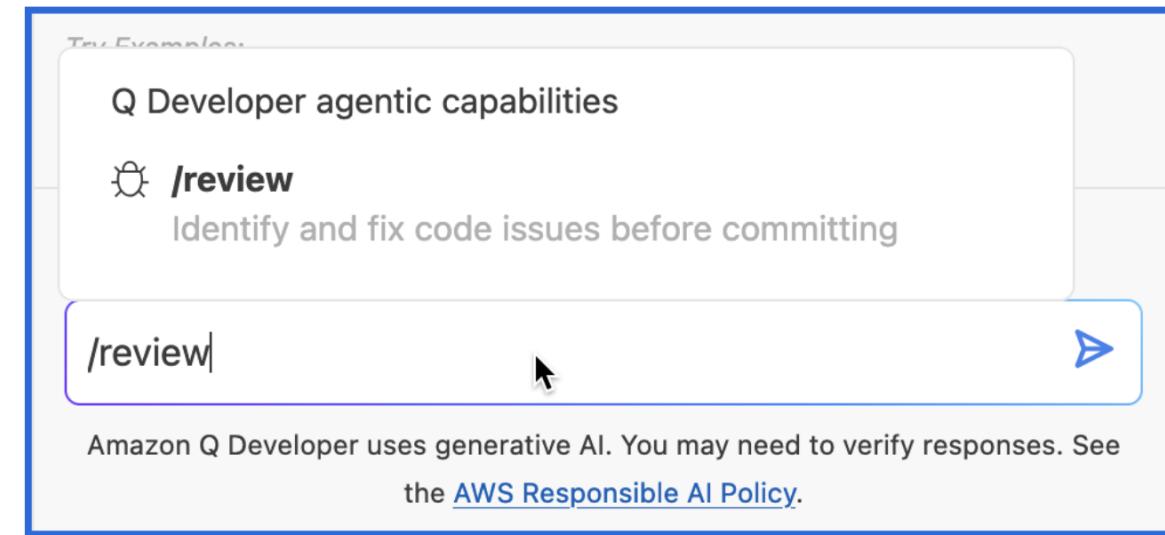
▶ AWS account access

▶ Content preferences

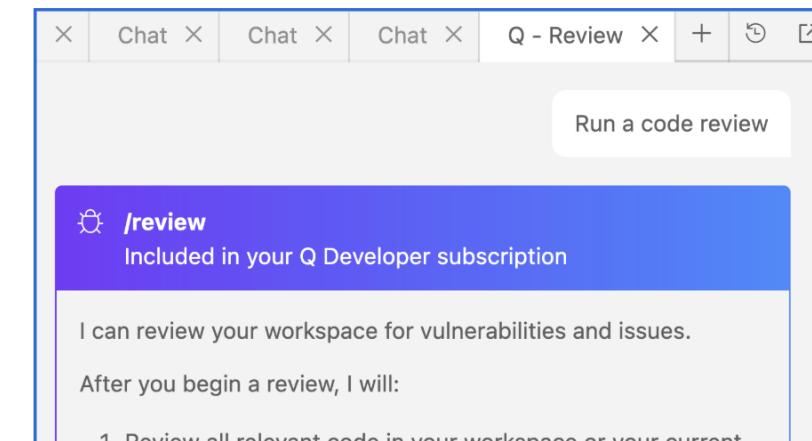
Exit event

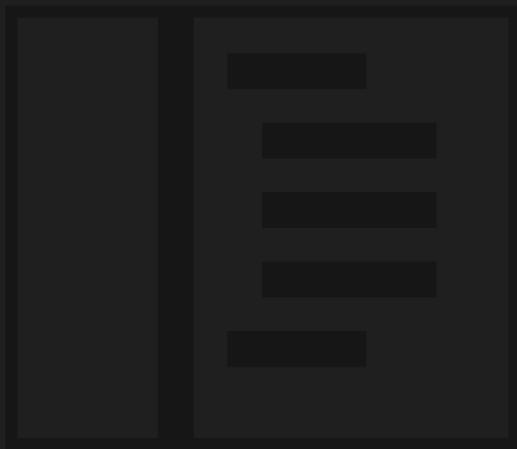
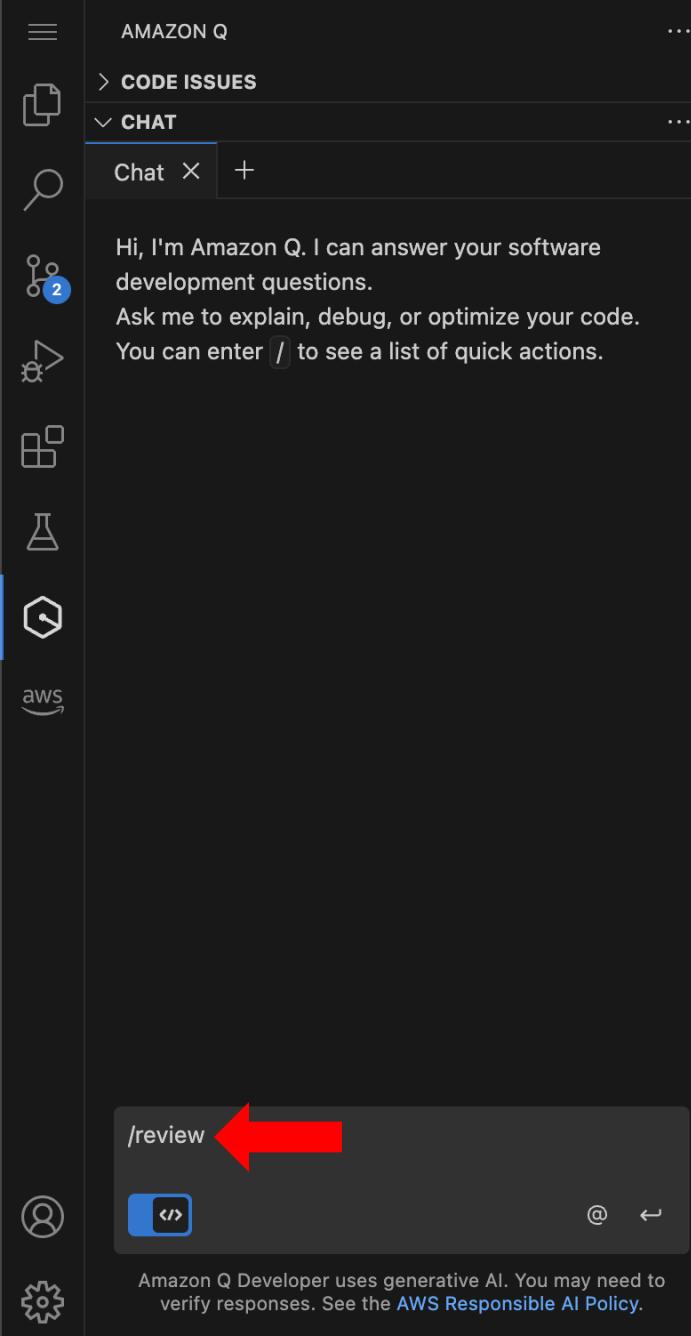
Activity 1: Running a Code Review

1. In the chat, type /review.



2. Follow the instructions in the Chat panel.





Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F



AMAZON Q

> CODE ISSUES

✓ CHAT

Turn OFF agentic coding

Run a code review

💡 /review
Included in your Q Developer subscription

I can review your workspace for vulnerabilities and issues.

After you begin a review, I will:

1. Review all relevant code in your workspace or your current file
2. Provide a list of issues for your review

You can then investigate, fix, or ignore issues.

To learn more, check out our [User Guide](#).

Would you like to review your active file or the workspace you have open?

←

Waiting on your inputs...

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

main* ↗ 0 ▲ 14 ⓘ 3 ⓘ 1 ✓ AWS: ec2:instance ▶ Amazon Q ⚡ Java: Ready

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS 1 CODE REFERENCE LOG

✖ Using tool: execute_bash (trusted)

⋮

● I will run the following shell command:
`cd /Workshop/java && for i in {1..5}; do curl -s "http://localhost:8090/game?user=test$i" > /dev/null; sleep 1; done`

⋮

↳ Purpose: Simulating multiple requests to test randomness

⋮

● Completed in 5.52s

⋮

Thinking...

AMAZON Q

...

CODE ISSUES

Critical 6 issues

- CWE-79,352,937,1035 - Package Vulnerability pom.xml [Ln 79, Col 1]
- CWE-94 - Unsanitized input is run as code GameController.java [Ln 33,...]
- CWE-94 - Unsanitized input is run as code GameController.java [Ln 56,...]
- CWE-94 - Unsanitized input is run as code HomeController.java [Ln 21,...]
- CWE-703 - Hardcoded credentials GameController.java [Ln 40, Col 1]

CHAT

Turn OFF agentic coding or share my progress here.

Initiating code review

Reviewing your code

Processing review results

...



I completed the code review. I found the following issues in your workspace:

- Critical: 6 issues
- High: 30 issues
- Medium: 13 issues
- Low: 7 issues
- Info: 0 issues



Ask a question or enter "/" for quick actions



Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F

Code Review Completed

Source: Amazon Q

View Code Issues

AMAZON Q

CODE ISSUES

C Critical 6 issues

- CWE-79,352,937,1035 - Package Vulnerability pom.xml [Ln ...]
- CWE-94 - Unsanitized input is run as code GameController.java [Ln 33, ...]
- CWE-94 - Unsanitized input is run as code GameController.java [Ln 56, ...]
- CWE-94 - Unsanitized input is run as code HomeController.java [Ln 21, ...]
- CWE-798 - Hardcoded credentials RandomUtils.java [Ln 19, Col 1]
- CWE-327,328,326,208,1240 - Insecure cryptography RandomUtils.jav...

H High 30 issues

- CWE-79 - Cross-site scripting GameController.java [Ln 47, Col 1]
- CWE-79 - Cross-site scripting GameController.java [Ln 70, Col 1]
- CWE-398 - Poor logging practice rule Word.java [Ln 62, Col 1]
- CWE-117,93 - Log injection Word.java [Ln 63, Col 1]
- CWE-398 - Poor logging practice rule Word.java [Ln 63, Col 1]
- CWE-398 - Poor logging practice rule Word.java [Ln 67, Col 1]
- CWE-117,93 - Log injection Word.java [Ln 67, Col 1]
- CWE-400 - Denial of service rule WordList.java [Ln 48, Col 1]
- CWE-398 - Poor logging practice rule WordList.java [Ln 55, Col 1]
- CWE-398 - Poor logging practice rule WordList.java [Ln 59, Col 1]
- CWE-117,93 - Log injection WordList.java [Ln 59, Col 1]
- Inadequate error handling detected. WordList.java [Ln 60, Col 1]
- CWE-398 - Poor logging practice rule WordList.java [Ln 63, Col 1]
- CWE-117,93 - Log injection WordList.java [Ln 63, Col 1]
- CWE-398 - Poor logging practice rule WordList.java [Ln 65, Col 1]
- CWE-117,93 - Log injection WordList.java [Ln 65, Col 1]
- CWE-398 - Poor logging practice rule WordList.java [Ln 72, Col 1]
- CWE-398 - Poor logging practice rule WordList.java [Ln 77, Col 1]
- CWE-398 - Poor logging practice rule WordList.java [Ln 79, Col 1]
- Inadequate error handling detected. WordList.java [Ln 107, Col 1]
- CWE-400 - Denial of service rule WordList.java [Ln 133, Col 1]
- CWE-329,1204,323 - Weak Initialization Vector (IV) WordList.java [Ln ...]
- Inadequate error handling detected. WordList.java [Ln 176, Col 1]

CHAT

Code Issue Details

CWE-79,352,937,1035 - Package Vulnerability Critical

A Cascading Style Sheets (CSS) injection vulnerability in Swagger UI before 3.23.11 allows attackers to use the Relative Path Overwrite (RPO) technique to perform CSS-based input field value exfiltration, such as exfiltration of a CSRF token value. In other words, this product intentionally allows the embedding of untrusted JSON data from remote servers, but it was not previously known that <style>@import within the JSON data was a functional attack method.

Relevant link: <https://github.com/advisories/GHSA-c427-hjc3-wrfw>

Common Weakness	Detector library	File path
Enumeration (CWE)	-	pom.xml [Ln 79]
CWE-79 , CWE-352 , CWE-937 , CWE-1035		

Suggested code fix preview

```
@@ -80,1 +80,1 @@
- <version>2.3.0</version>
+ <version>2.10.0</version>
```

[Open diff](#) [Copy](#)

Why are we recommending this?

Update installed software package springfox-swagger-ui to version 2.10.0

Accept Fix Regenerate Fix Explain Ignore Ignore All

main* AWS: ec2:instance Amazon Q Java: Ready Layout: U.S.

AMAZON Q

CODE ISSUES

- Critical 6 issues
 - CWE-79,352,937,1035 - Package Vulnerability pom.xml [Ln ...]
 - CWE-94 - Unsanitized input is run as code GameController.java [Ln 33, ...]
 - CWE-94 - Unsanitized input is run as code GameController.java [Ln 56, ...]
 - CWE-94 - Unsanitized input is run as code HomeController.java [Ln 21, ...]
 - CWE-798 - Hardcoded credentials RandomUtils.java [Ln 19, Col 1]
 - CWE-327,328,326,208,1240 - Insecure cryptography RandomUtils.java...
- High 30 issues
 - CWE-79 - Cross-site scripting GameController.java [Ln 47, Col 1]
 - CWE-79 - Cross-site scripting GameController.java [Ln 70, Col 1]
 - CWE-398 - Poor logging practice rule Word.java [Ln 62, Col 1]
 - CWE-117,93 - Log injection Word.java [Ln 63, Col 1]
 - CWE-398 - Poor logging practice rule Word.java [Ln 63, Col 1]
 - CWE-398 - Poor logging practice rule Word.java [Ln 67, Col 1]
 - CWE-117,93 - Log injection Word.java [Ln 67, Col 1]
 - CWE-400 - Denial of service rule WordList.java [Ln 48, Col 1]
 - CWE-398 - Poor logging practice rule WordList.java [Ln 55, Col 1]
 - CWE-398 - Poor logging practice rule WordList.java [Ln 59, Col 1]
 - CWE-117,93 - Log injection WordList.java [Ln 59, Col 1]
 - Inadequate error handling detected. WordList.java [Ln 60, Col 1]
 - CWE-398 - Poor logging practice rule WordList.java [Ln 63, Col 1]
 - CWE-117,93 - Log injection WordList.java [Ln 63, Col 1]
 - CWE-398 - Poor logging practice rule WordList.java [Ln 65, Col 1]
 - CWE-117,93 - Log injection WordList.java [Ln 65, Col 1]
 - CWE-398 - Poor logging practice rule WordList.java [Ln 72, Col 1]
 - CWE-398 - Poor logging practice rule WordList.java [Ln 77, Col 1]
 - CWE-398 - Poor logging practice rule WordList.java [Ln 79, Col 1]
 - Inadequate error handling detected. WordList.java [Ln 107, Col 1]
 - CWE-400 - Denial of service rule WordList.java [Ln 133, Col 1]
 - CWE-329,1204,323 - Weak Initialization Vector (IV) WordList.java [Ln ...]
 - Inadequate error handling detected. WordList.java [Ln 176, Col 1]

CHAT

java

CWE-79,352,937,1035 - Package Vulnerability Critical

A Cascading Style Sheets (CSS) injection vulnerability in Swagger UI before 3.23.11 allows attackers to use the Relative Path Overwrite (RPO) technique to perform CSS-based input field value exfiltration, such as exfiltration of a CSRF token value. In other words, this product intentionally allows the embedding of untrusted JSON data from remote servers, but it was not previously known that <style>@import within the JSON data was a functional attack method.

Relevant link: <https://github.com/advisories/GHSA-c427-hjc3-wrfw>

Common Weakness	Detector library	File path
Enumeration (CWE)	-	pom.xml [Ln 79]
CWE-79 , CWE-352 , CWE-937 , CWE-1035		

Suggested code fix preview

```
@@ -80,1 +80,1 @@
- <version>2.3.0</version>
+ <version>2.10.0</version>
```

[Open diff](#) [Copy](#)

Why are we recommending this?

Update installed software package springfox-swagger-ui to version 2.10.0

Accept Fix Generate Fix Explain Ignore Ignore All

main* AWS: ec2:instance Amazon Q Java: Ready Layout: U.S.

AMAZON Q

CODE ISSUES

- C Critical 5 issues
 - CWE-94 - Unsanitized input is run as code GameController.java [Ln 33, Col 1]
 - CWE-94 - Unsanitized input is run as code GameController.java [Ln 56, Col 1]
 - CWE-94 - Unsanitized input is run as code HomeController.java [Ln 21, Col 1]
 - CWE-798 - Hardcoded credentials RandomUtils.java [Ln 19, Col 1]
 - CWE-327,328,326,208,1240 - Insecure cryptography RandomUtils.java [Ln 19, Col 1]
- H High 30 issues
 - CWE-79 - Cross-site scripting GameController.java [Ln 47, Col 1]
 - CWE-79 - Cross-site scripting GameController.java [Ln 70, Col 1]
 - CWE-398 - Poor logging practice rule Word.java [Ln 62, Col 1]
 - CWE-117,93 - Log injection Word.java [Ln 63, Col 1]
 - CWE-398 - Poor logging practice rule Word.java [Ln 63, Col 1]
 - CWE-398 - Poor logging practice rule Word.java [Ln 67, Col 1]
 - CWE-117,93 - Log injection Word.java [Ln 67, Col 1]
 - CWE-400 - Denial of service rule WordList.java [Ln 48, Col 1]
 - CWE-398 - Poor logging practice rule WordList.java [Ln 55, Col 1]
 - CWE-398 - Poor logging practice rule WordList.java [Ln 59, Col 1]
 - CWE-117,93 - Log injection WordList.java [Ln 59, Col 1]
 - Inadequate error handling detected. WordList.java [Ln 60, Col 1]
 - CWE-398 - Poor logging practice rule WordList.java [Ln 63, Col 1]
 - CWE-117,93 - Log injection WordList.java [Ln 63, Col 1]
 - CWE-398 - Poor logging practice rule WordList.java [Ln 65, Col 1]
 - CWE-117,93 - Log injection WordList.java [Ln 65, Col 1]
 - CWE-398 - Poor logging practice rule WordList.java [Ln 72, Col 1]
 - CWE-398 - Poor logging practice rule WordList.java [Ln 77, Col 1]
 - CWE-398 - Poor logging practice rule WordList.java [Ln 79, Col 1]
 - Inadequate error handling detected. WordList.java [Ln 107, Col 1]
 - CWE-400 - Denial of service rule WordList.java [Ln 133, Col 1]
 - CWE-329,1204,323 - Weak Initialization Vector (IV) WordList.java [Ln 133, Col 1]
 - Inadequate error handling detected. WordList.java [Ln 176, Col 1]
 - CWE-522 - Not recommended aws credentials classes RandomUtils.java [Ln 19, Col 1]

CHAT

pom.xml M X

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.2</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.amazon</groupId>
    <artifactId>QWordsService</artifactId>
    <version>0.0.1</version>
    <name>QWordsService</name>
    <description>Word guessing game application implemented in a springboot MVC architecture</description>
    <properties>
        <java.version>1.8</java.version>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
        <aws.java.sdk.version>1.12.405</aws.java.sdk.version>
    </properties>
    <dependencyManagement>
        <dependencies>
            <dependency>
                <groupId>com.amazonaws</groupId>
                <artifactId>aws-java-sdk-kinesis</artifactId>
                <version>${aws.java.sdk.version}</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>
```

A build file was modified. Do you want to synchronize the Java classpath/configuration?

Source: Language Support for Java(TM) SE API ... Yes Never

main* 0 69 4 AWS: ec2:instance Amazon Q Java: Ready

Ln 1, Col 1 Tab Size: 4 UTF-8 LF {} XML Layout: U.S.

The screenshot shows a Java development environment with the following interface elements:

- Left Sidebar:** Contains icons for file operations, search, navigation, and AWS services (Amazon Q, AWS Lambda, CloudWatch Metrics).
- Top Bar:** Includes a back/forward button, a search bar with the query "java", and system status indicators.
- Central Area:** A tree view titled "CODE ISSUES" showing two categories:
 - Critical:** 5 issues (e.g., CWE-94, CWE-798, CWE-327)
 - High:** 30 issues (e.g., CWE-79, CWE-398, CWE-117, CWE-400)
- Code Editor:** An open file named "pom.xml" with the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.2</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.amazon</groupId>
    <artifactId>QWordsService</artifactId>
    <version>0.0.1</version>
    <name>QWordsService</name>
    <description>Word guessing game application implemented in a springboot MVC architecture</description>
    <properties>
        <java.version>1.8</java.version>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
        <aws.java.sdk.version>1.12.405</aws.java.sdk.version>
    </properties>
    <dependencyManagement>
        <dependencies>
            <dependency>
                <groupId>com.amazonaws</groupId>
                <artifactId>aws-java-sdk-bom</artifactId>
                <version>${aws.java.sdk.version}</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>

```
- Right Sidebar:** A vertical stack of code snippets or logs.
- Bottom Bar:** Includes tabs for "main*", "AWS: ec2:instance", "Amazon Q", "Java: Ready", and a status bar with "Ln 1, Col 1" and "Tab Size: 4".

Add a Feature



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

▼ Code Comprehension and Summarization

Task 1: Learn about Amazon Q

Task 2: Learn About the Application Structure

Task 3: Create documentation

▼ Code Debugging, Testing, and Optimization

Task 1: Run and Play Q-Words

Task 2: Identify and Fix a Bug with Amazon Q

Task 3: Writing Unit Tests with Amazon Q Developer

Task 4: Review your code with Amazon Q Developer

▼ Natural Language to New Feature

Task 1: Describe the Feature to Amazon Q

Task 2: Crafting Effective Prompts for Amazon Q Developer

Task 3: Migrating Between Programming Languages

► Application Modernization and

► AWS account access

► Content preferences

Exit event

- Write a prompt similar to the one below, I'd like to be able to see the history of guesses and the corresponding guess information for each character. This will help the player avoid repeating the same guesses while remembering the feedback about each character in the guess.

Hi, I'm Amazon Q.
Where would you like to start?

Feature development
Amazon Q Developer Agent for Software Development

After you provide a task, I will:

1. Generate code based on your description and the code in your workspace
2. Provide a list of suggestions for you to review and add to your workspace
3. If needed, iterate based on your feedback To learn more, visit the [user guide](#)

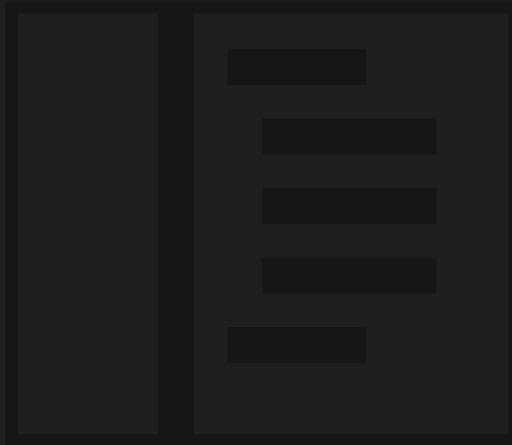
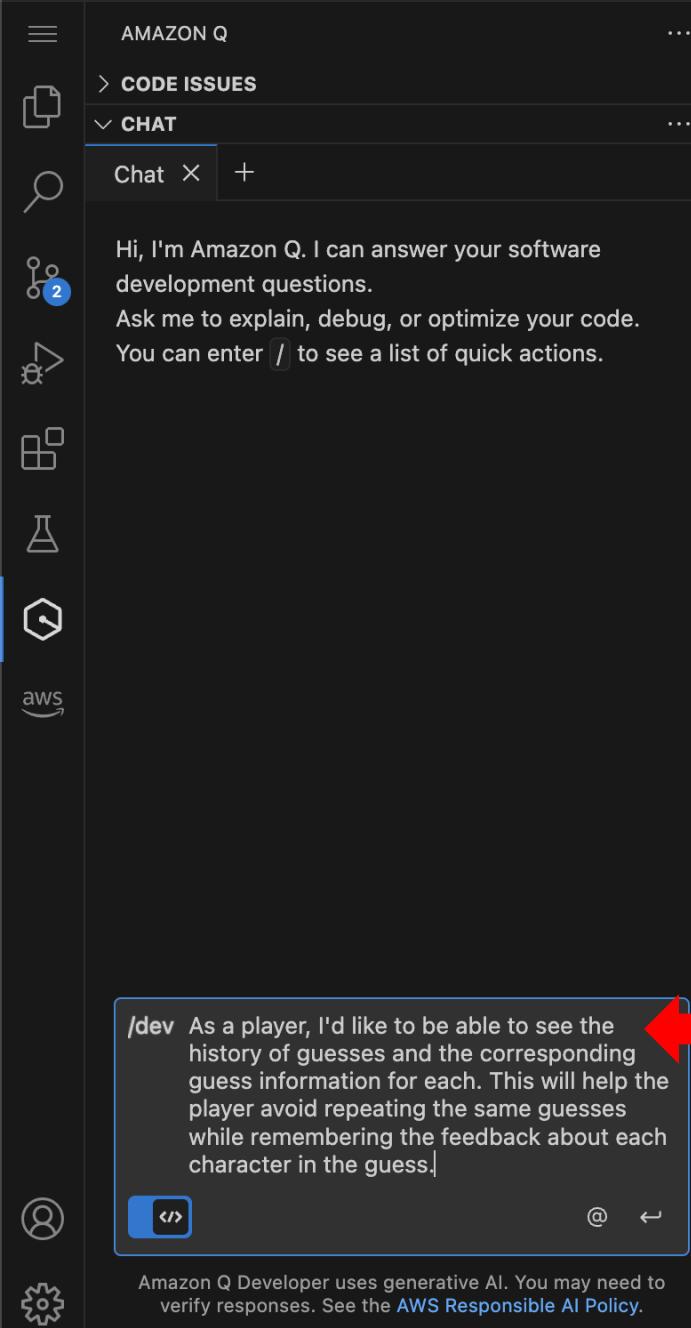
As a player, I'd like to be able to see the history of guesses and the corresponding guess information for each. This will help the player avoid repeating the same guesses while remembering the feedback about each character in the guess.

Okay, I'll generate code for that.

This might take a few minutes. You can navigate away from this chat, but please keep this tab open. I'll notify you when I'm done.

Generating code...





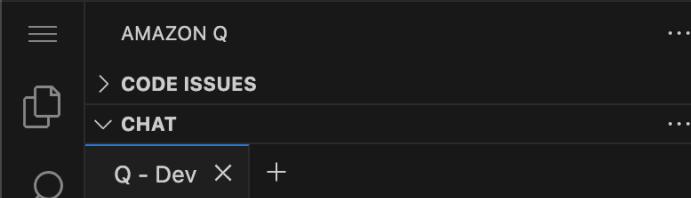
Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F





As a player, I'd like to be able to see the history of guesses and the corresponding guess information for each. This will help the player avoid repeating the same guesses while remembering the feedback about each character in the guess.

💡 Feature development
Amazon Q Developer Agent for Software Development

After you provide a task, I will:

1. Generate code based on your description and the code in your workspace
2. Provide a list of suggestions for you to review and add to your workspace
3. If needed, iterate based on your feedback

To learn more, visit the [user guide](#)

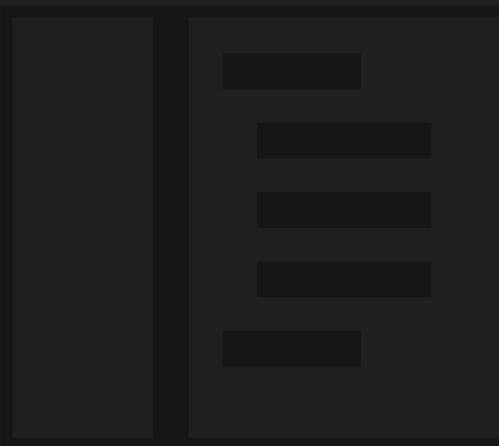
Okay, I'll generate code for that.



Uploading code...



Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F



AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Q - Dev X +

Generating code...

I'll share updates here as I work on this.

Summary of changes

1. First, I'll explore the repository structure to understand the codebase and locate relevant files. I'll start by examining the source code directory.
2. Let me check the Java source files to find the game-related classes.
3. I'll examine the GameController and GameStatus classes first, as they are likely the most relevant for implementing guess history functionality.

Files used

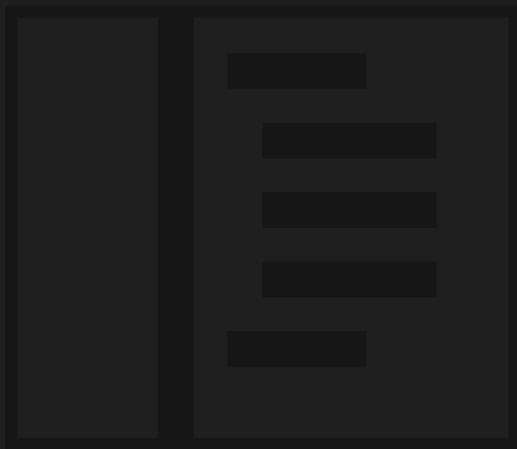
These fields can change throughout code generation

- Reviewed: 6
 - src/main
 - src/main/java/com
 - src/main/java/com/sample/qwords
 - src/main/java/com/sample/qwords/controller /GameController.java
 - src/main/java/com/sample/qwords/model/Ga

Generating code...

■ Stop

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F



AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Q - Dev X +

Generating code...

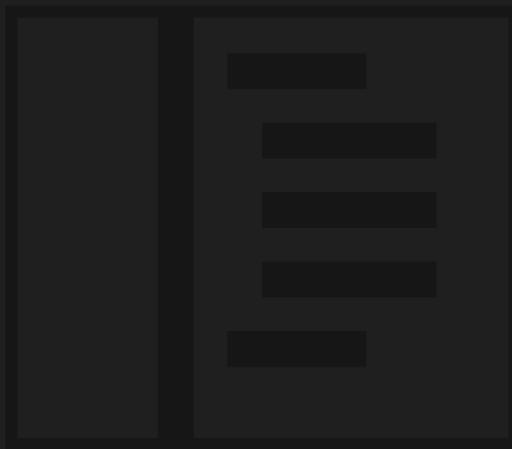
I'll share updates here as I work on this.

Summary of changes

1. First, I'll explore the repository structure to understand the codebase and locate relevant files. I'll start by examining the source code directory.
2. Let me check the Java source files to find the game-related classes.
3. I'll examine the GameController and GameStatus classes first, as they are likely the most relevant for implementing guess history functionality.
4. I need to create a GuessHistory class to store the history of guesses and their feedback. Then I'll modify the GameController to maintain this history.
5. Now I'll modify the GameController to use GuessHistory and maintain the history of guesses in the session.
6. I apologize for the error. Let me try again:
7. Now let's modify the index method to initialize the guess history:
8. Now let's update the makeGuess method to

Choose an option to proceed

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F



AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Q - Dev X +

8. Now let's update the makeGuess method to store the guess history:

9. Now I need to add the necessary imports for the GuessHistory class in GameController.

10. Now I need to check the game template to add the display of guess history. Let me look for the template file.

11. I'll add a new section to display the guess history in the game template, showing each guess and its corresponding feedback.

12. Now I need to create unit tests for the GuessHistory class and update the GameController tests to include the new functionality.

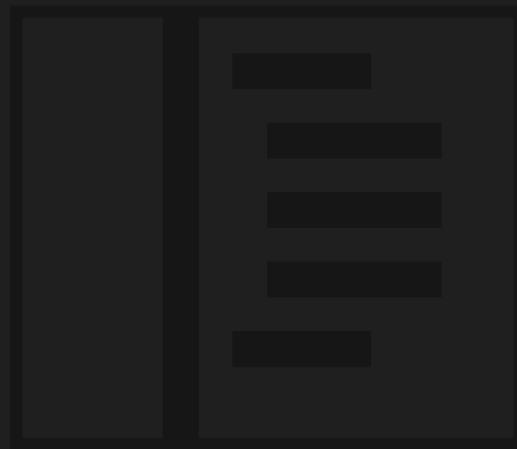
13. Let's create a test directory for the model package and add tests for GuessHistory:

14. Now let's create a test for the GameController to verify the guess history functionality:

15. I have completed the implementation of the guess history feature. Here's a summary of the changes made: 1. Created GuessHistory class to store guess attempts and their feedback 2. Modified GameController to maintain guess history in the session 3. Updated the game template to display the guess history 4. Added

Choose an option to proceed

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



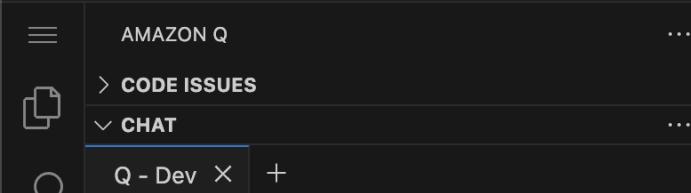
Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F





15. I have completed the implementation of the guess history feature. Here's a summary of the changes made: 1. Created GuessHistory class to store guess attempts and their feedback 2. Modified GameController to maintain guess history in the session 3. Updated the game template to display the guess history 4. Added unit tests for both GuessHistory and the updated GameController. The feature now allows players to see their previous guesses and the corresponding feedback, helping them avoid repeating guesses and remember character feedback.

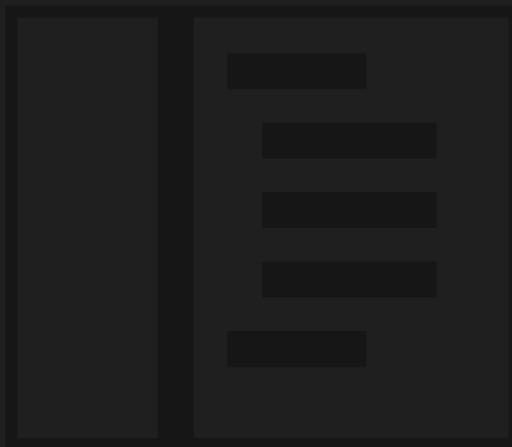
Files used

These fields can change throughout code generation

- Reviewed: 11
 - src/main
 - src/main/java/com
 - src/main/java/com/sample/qwords
 - src/main/java/com/sample/qwords/controller/GameController.java
 - src/main/java/com/sample/qwords/model/GameStatus.java

Choose an option to proceed

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F



AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

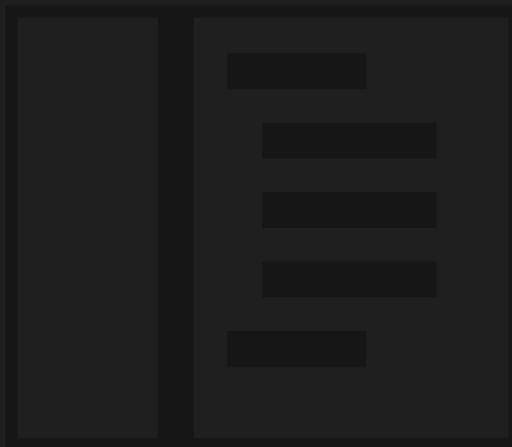
Q - Dev X +

- Created: 3
 - src/main/java/com/sample/qwords/model/GuessHistory.java
 - src/test/java/com/sample/qwords/model/GuessHistoryTest.java
 - src/test/java/com/sample/qwords/controller/GameControllerTest.java
- Changed: 2
 - src/main/java/com/sample/qwords/controller/GameController.java
 - src/main/resources/templates/game.html
- Deleted: 0

Code suggestions 5 file(s)

✓ □ Changes 1 file(s)

- ✓ □ src 2 file(s)
 - ✓ □ main 2 file(s)
 - ✓ □ java 1 file(s)
 - ✓ □ com 1 file(s)
 - ✓ □ sample 1 file(s)



Show All Commands ⌘ P

Go to File ⌘ P

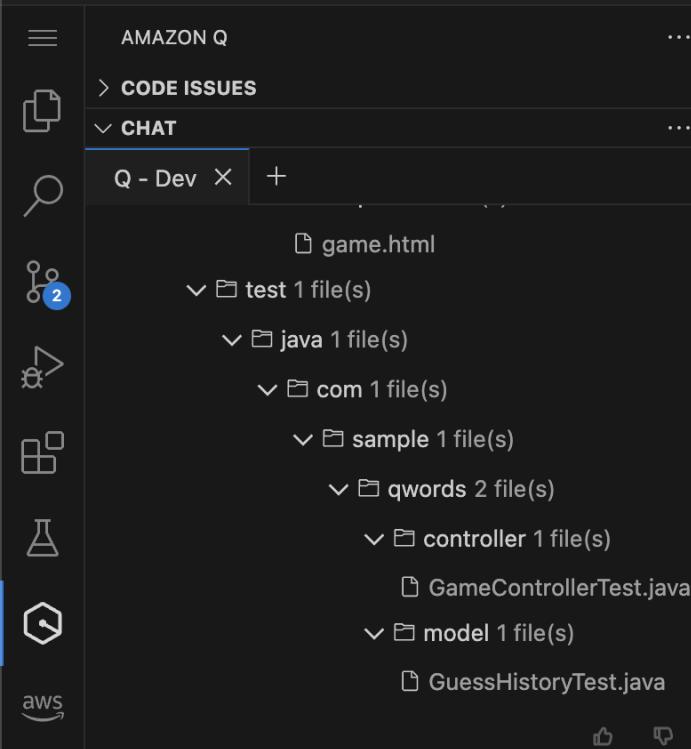
Open Settings ⌘ ,

Find in Files ⌘ F

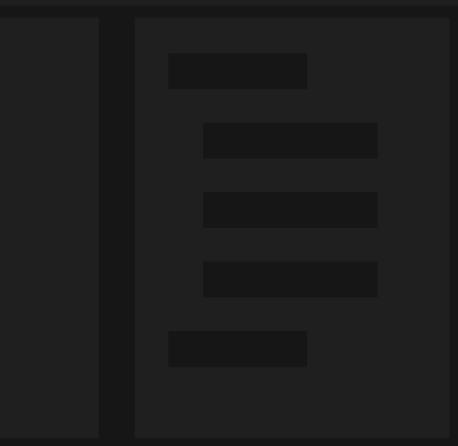
Choose an option to proceed

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).





Would you like me to add this code to your project,
or provide feedback for new code?



Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F

Choose an option to proceed

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Q - Dev X +

- ✓ templates 1 file(s)
 - ✓ game.html Change accepted
- ✓ test 1 file(s)
 - ✓ java 1 file(s)
 - ✓ com 1 file(s)
 - ✓ sample 1 file(s)
 - ✓ qwords 2 file(s)
 - ✓ controller 1 file(s)
 - ✓ GameControllerTest.java
 - ✓ model 1 file(s)
 - ✓ GuessHistoryTest.java C



Would you like me to add this code to your project,
or provide feedback for new code?

Okay, I updated your code files. Would you like to
work on another task?



Choose an option to proceed



Amazon Q Developer uses generative AI. You may need to
verify responses. See the [AWS Responsible AI Policy](#).



Show All Commands ⌘ P

Go to File ⌘ P

Open Settings ⌘ ,

Find in Files ⌘ F



AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Q - Dev X +

- ✓ templates 1 file(s)
 - ✓ game.html Change accepted
- ✓ test 1 file(s)
 - ✓ java 1 file(s)
- ✓ com 1 file(s)
 - ✓ sample 1 file(s)
 - ✓ qwords 2 file(s)
 - ✓ controller 1 file(s)
 - ✓ GameControllerTest.java
 - ✓ model 1 file(s)
 - ✓ GuessHistoryTest.java C



Would you like me to add this code to your project,
or provide feedback for new code?

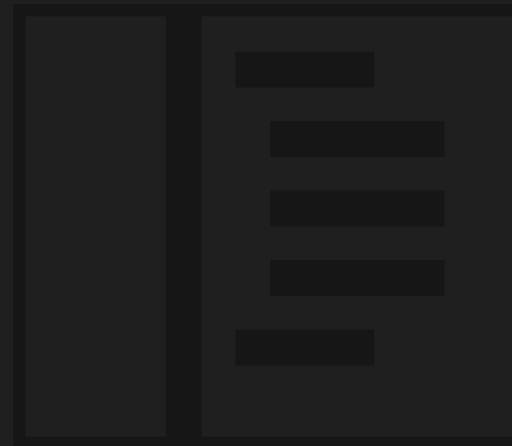
Okay, I updated your code files. Would you like to
work on another task?



Choose an option to proceed



Amazon Q Developer uses generative AI. You may need to
verify responses. See the [AWS Responsible AI Policy](#).



PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

q + v ⊞ ⊖ ⊖ ^ x

Word: butane

Guess: animal

Evaluating current char from guess to word: a->b
Evaluating current char from guess to word: n->u
Evaluating current char from guess to word: i->t
Evaluating current char from guess to word: m->a
Evaluating current char from guess to word: a->n
Evaluating current char from guess to word: l->e
2025-05-31 17:16:58.632 INFO 28622 --- [io-8090-exec-10] c.s.qwords.controller.GameController : User Michael
el made a guess: animal
2025-05-31 17:16:58.632 INFO 28622 --- [io-8090-exec-10] c.s.qwords.controller.GameController : Guess: ani
mal, model: {word=butane, attempts=1}, user: Michael
^C

> Can you run this Spring Boot application for me? 



AMAZON Q ...

> CODE ISSUES

✓ CHAT ...

Q - Dev X +

- ✓ templates 1 file(s)
 - ✓ game.html Change accepted
- ✓ test 1 file(s)
 - ✓ java 1 file(s)
- ✓ com 1 file(s)
 - ✓ sample 1 file(s)
 - ✓ qwords 2 file(s)
 - ✓ controller 1 file(s)
 - ✓ GameControllerTest.java
 - ✓ model 1 file(s)
 - ✓ GuessHistoryTest.java C

Would you like me to add this code to your project, or provide feedback for new code?

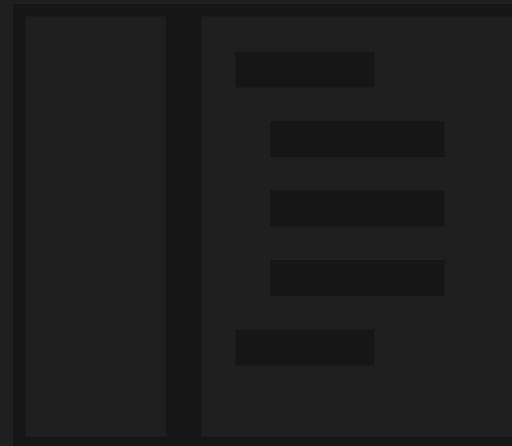
Okay, I updated your code files. Would you like to work on another task?



Choose an option to proceed



Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).



PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS 1 CODE REFERENCE LOG

index: 9
Total words: 37
Words loaded from classpath resource: words.txt
2025-05-31 17:28:30.919 INFO 34166 --- [
index: 0
2025-05-31 17:28:31.235 INFO 34166 --- [
 endpoint(s) beneath base path '/actuator'
2025-05-31 17:28:31.259 INFO 34166 --- [
 rted on port(s): 8090 (http) with context path ''
2025-05-31 17:28:31.267 INFO 34166 --- [
 refreshed
2025-05-31 17:28:31.274 INFO 34166 --- [
 custom documentation plugin(s)
2025-05-31 17:28:31.278 INFO 34166 --- [
 or api listing references
2025-05-31 17:28:31.284 INFO 34166 --- [
 ordsApplication in 1.363 seconds (JVM running for 1.52)

main] com.sample.qwords.repository.WordList : Selected i
main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 1
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat sta
main] d.s.w.p.DocumentationPluginsBootstrapper : Context re
main] d.s.w.p.DocumentationPluginsBootstrapper : Found 101
main] s.d. ⓘ Your application running on port 8090 is available. See all ↗ forwarded ports
main] com.

Open in Browser





Words

Please enter your name:

Play



Words

Please enter your name:

Play



Words

Make your first guess!

Enter a 6-letter word.

MAKE GUESS

Attempts: 0

Enter a word with 6 letters



Words

Try again!

Enter a 6-letter word.

MAKE GUESS

asdfgh

Result:



Attempts: 1

Guess History:





Words

Try again!

Enter a 6-letter word.

MAKE GUESS

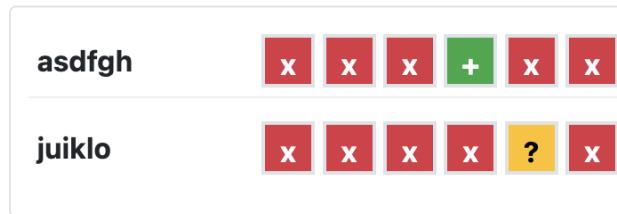
juiklo

Result:



Attempts: 2

Guess History:





Words

Try again!

Enter a 6-letter word.

MAKE GUESS

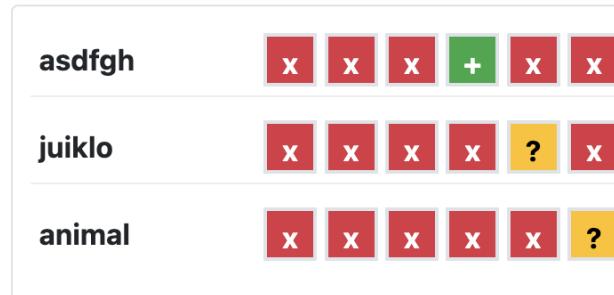
animal

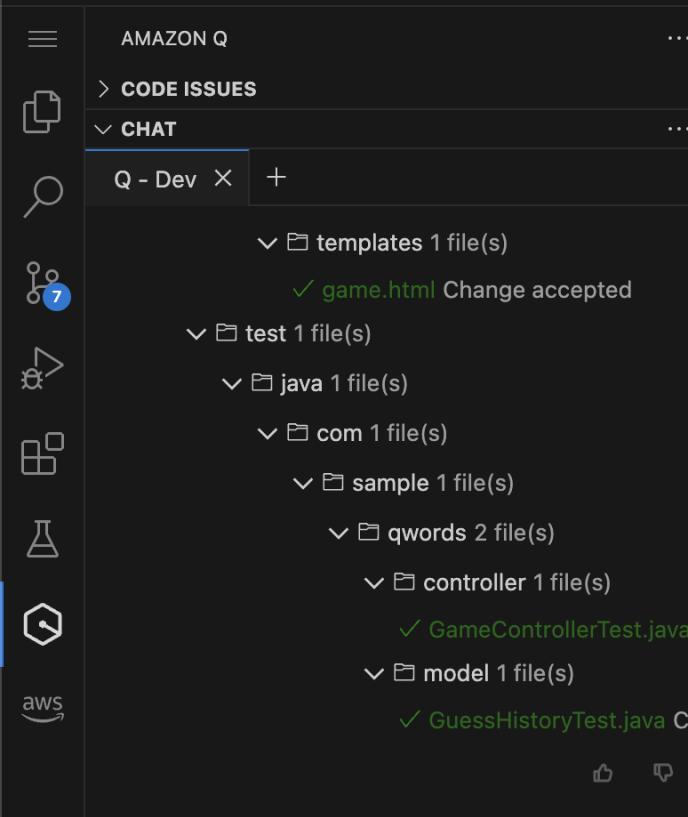
Result:



Attempts: 3

Guess History:





Would you like me to add this code to your project, or provide feedback for new code?

Okay, I updated your code files. Would you like to work on another task?

Choose an option to proceed

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS 1 CODE REFERENCE LOG

Word: belfry

Guess: animal

Evaluating current char from guess to word: a->b
Evaluating current char from guess to word: n->e
Evaluating current char from guess to word: i->l
Evaluating current char from guess to word: m->f
Evaluating current char from guess to word: a->r
Evaluating current char from guess to word: l->y

2025-05-31 17:29:24.757 INFO 34166 --- [nio-8090-exec-1] c.s.qwords.controller.GameController : User Michael made a guess: animal
2025-05-31 17:29:24.757 INFO 34166 --- [nio-8090-exec-1] c.s.qwords.controller.GameController : Guess: animal, model: {guessHistory=com.sample.qwords.model.GuessHistory@ce2e2b3, word=belfry, attempts=2}, user: Michael

^C





Thank you!

Michael Lin

linmicht@amazon.com