

Read OTP from Gmail using Java + Google OAuth2

Automate secure email reading for testing workflows

By Shashank Awasthi

Why Read OTP from Gmail?

- Automate 2FA flows
- Fetch real-time OTPs
- Ideal for test automation

By Shashank Awasthi

OAuth2 Flow in a Nutshell

1. App requests Gmail access
2. User consents
3. App receives token
4. Gmail API fetches email

By Shashank Awasthi

Step 1: Google Cloud Setup

- Go to `console.cloud.google.com`
- Create project
- Enable Gmail API
- Setup OAuth Consent Screen

By Shashank Awasthi

Step 2: Credentials Setup

- Create OAuth Client ID
- Type: Desktop
- Download `credentials.json`
- Place in project folder

By Shashank Awasthi

Step 3: Maven Dependencies

```
<dependency>
  <groupId>com.google.api-client</groupId>
  <artifactId>google-api-client</artifactId>
  <version>1.34.1</version>
</dependency>
<dependency>
  <groupId>com.google.apis</groupId>
  <artifactId>google-api-services-gmail</artifactId>
  <version>v1-rev110-1.25.0</version>
</dependency>
```

By Shashank Awasthi

Step 4: Gmail Auth Utility

```
GoogleClientSecrets secrets = GoogleClientSecrets.load(...);
GoogleAuthorizationCodeFlow flow = new GoogleAuthorizationCodeFlow.Builder(...).build();
Credential credential = new AuthorizationCodeInstalledApp(flow, ...).authorize("user");
Gmail service = new Gmail.Builder(...).build();
```

By Shashank Awasthi

Step 5: Read OTP Email

```
String query = "subject:OTP is:unread";
ListMessagesResponse res = service.users().messages().list("me").setQ(query).execute();
Message msg = service.users().messages().get("me",
res.getMessages().get(0).getId()).execute();
String otp = msg.getSnippet().replaceAll("[^0-9]", "").substring(0, 6);
```

By Shashank Awasthi

Output + Use Case

- OTP Received: 837261
- Used in automated test
- Marked email as read

By Shashank Awasthi

What's Next?

- Archive email
- Push to GitHub
- Integrate into Selenium or TestNG framework

By Shashank Awasthi