Spendgo Identity for Mobile

#mobile

Integration Use Cases

The following is best for integrating Spendgo into:

- Native mobile applications

Endpoints

When enabling Spendgo Identity in a mobile application, these are the API endpoints that are most commonly used:

POST /mobile/gen/<WEB_CONTEXT>/v1/lookup
Retrieves a member status

POST /mobile/gen/<WEB_CONTEXT>/v1/get
Retrieves the member details

POST /mobile/gen/<WEB_CONTEXT>/v1/add
Creates a new member

POST /mobile/gen/<WEB_CONTEXT>/v1/signin
Creates a member access token and refresh token

POST /mobile/gen/<WEB_CONTEXT>/v1/forgotpassword
Triggers a password reset workflow for a member

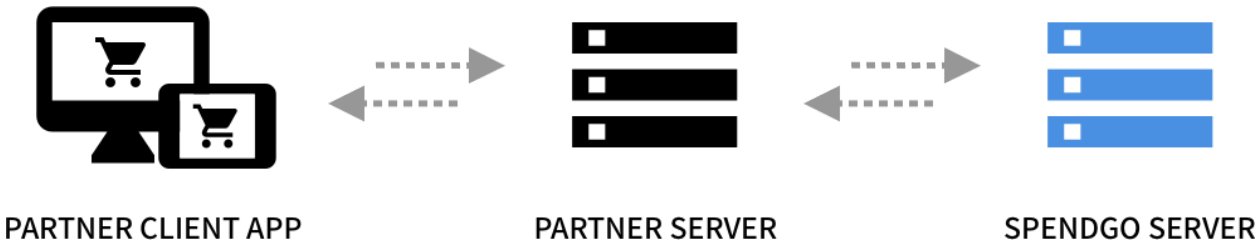
POST /mobile/gen/<WEB_CONTEXT>/v1/update
Updates a member.

POST /mobile/gen/<WEB_CONTEXT>/v1/signoff
Creates a member log out.

See API Reference document.

Security Requirements

The mobile APIs use a combination of signed requests and an authentication token for authentication. These are deemed to be server-to-server integrations and the following diagram depicts the typical data flow between the Partner and Spendgo systems:



Some APIs – Retrieve a member status, Create a member, and Retrieve stores – have anonymous access that only require a signed request (using the provided client ID) and does not require the member to sign in.

Getting Your Credentials

You use different types of security credentials depending on how you interact with Spendgo. For signed requests, Spendgo will share with you the following credentials:

- API Key
- Shared Client Secret
- Web Context

Note: *Security credentials are environment and account-specific. If you have access to multiple Spendgo accounts, use the credentials that are associated with the account and environment that you want to access.*

Customer Experience

A typical customer experience may look like the following:

If you are also building a mobile loyalty experience, once a member has been authenticated, you may use the authentication token for the server-to-server Order API requests.

1. Look up a member

To create a member on a mobile device, you must create a form with the minimum profile fields to collect:

- Phone
- Email
- Password

It is also required to Include a terms and conditions caption with links next to the form submission button.

Example

```
By clicking on “Submit” you agree to the Spendgo Terms and Privacy Policy.
```

When a member enters their phone or email into the create new member form, you will use the Retrieve a member status request to validate the phone and email (in separate calls) do not already exist and are unique.

HTTP request

```
POST https://my.<ENVIRONMENT>.com/mobile/gen/<WEB_CONTEXT>/v1/lookup
```

Based on the `status` / `linked_to_account` response combinations, the following are common user experience actions:

- `Activated` / `true` is a registered member with your account, direct member to sign in.
- `Activated` / `false` is a registered member in the Spendgo system associated with other accounts, but is not associated with the account your request is targeting. Prompt user to sign in, this will automatically link the member to your account.
- `StarterAccount` / `true` is a starter member with your account. Direct the member to sign up and register their account.
- `StarterAccount` / `false` is a starter member in the Spendgo system associated with other accounts, but is not associated with the account your request is targeting. Direct the member to sign up and register their account.
- `NotFound` is a phone or email that does not exist in the Spendgo system. Direct the user to sign up and register their account.
- `InvalidEmail` is a non-valid email format entered. Prompt the user an error message to check their entry.

2. Create a member

Once you have confirmed the member phone and email do not already exist, a new member account can be created. Your application server will use the Create a member request:

HTTP request

```
POST https://my.<ENVIRONMENT>.com/mobile/gen/<WEB_CONTEXT>/v1/add
```

- Once the new member request is posted, the member will be sent a verification link to the provided email. The user is required to verify their email address to complete their registration and become a member.

3. Member sign in

The mobile application retrieves the user credentials – phone or email and password – directly and using the Sign in member request, your app will validate the credentials and generate a mobile access token and refresh token used for subsequent calls to other secure APIs.

As a matter of security, the credentials read from the user should never be stored locally within the mobile application. It should only be used for the duration of the sign in request.

HTTP request

```
POST https://my.<ENVIRONMENT>.com/mobile/gen/<WEB_CONTEXT>/v1/signin
```

Returns

Returns a token and `spendgo_id` on a `200` successful request. When the member is not found, a `404` response is returned. When the user id and/or password are invalid, a `401` response is returned.

When a member has forgotten their password, your mobile app can use the Reset member password request to initiate the reset password flow, which begins with a reset password link sent to their email. The reset experience is handled on the Spendgo servers.

4. Access customer information using the access token

After your mobile application obtains a mobile token it can be used to make further subsequent calls to Spendgo's secure API to fetch and manage the member profile and loyalty information. The following are APIs for mobile use:

- Refresh Token
- Retrieve a member
- Update a member
- Retrieve member balance
- Orders (these are server-to-server requests)
- Sign out member

Related Documents

- Add Transaction Using Signed Requests
- API Reference