Table of Contents

Single Sign-On (SSO)

Provisioning and Deprovisioning
  Option 1 - Just-in-time (JIT)
  Option 2 - SCIM

Grammarly Business for Windows
  Deploying the Grammarly Business for Windows
    Option 1 - Using Intune
    Option 2 - Using SCCM

Grammarly Business for Mac OS
  Deploying the Grammarly Business for Mac
    Using JAMF (Script Method)

Managed Mode
  What is Managed Mode?
  Enabling Managed Mode
Single Sign-On (SSO)

As an administrator of a Grammarly Business account with over 50 seats, you can set up SAML single sign-on for your team:

**Step 1 - Configure Grammarly on your Identity Provider**

We currently have connectors with the following IdPs, which you can find in their application catalog:

- OneLogin
- Okta
- Azure AD
- PingIdentity

For all other Identity Providers, you can configure a custom SAML application with the following parameters:

1. Single sign-on URL / ACS URL:  
   `https://sso.grammarly.com/saml/assertion`
2. Audience URI/SP Entity ID/Issuer:
   `https://sso.grammarly.com/saml/metadata`
3. Name ID format: **EmailAddress**
4. Application username: **Email**
5. Attributes/Claims:
   - EmailAddress
   - FirstName
   - LastName
Step 2 - Configure your Identity Provider in Grammarly

1. Log in to your Grammarly account, go to the Authentication page, and click **Configure**.

2. Fill in the necessary fields. The required information is available in your IdP’s portal. Please see this page for more details.
3. Click **Test SSO**. If successful, click **Activate SSO** to set up this feature for your team.

**Note:** If the SSO test has failed, please double-check the Grammarly settings in your IdP as well as the values you entered on the Authentication page.

After SSO has been set up, Grammarly will rely on your identity provider to manage users on your Grammarly Business team.

Users who are already on your team won’t be able to log in to Grammarly until you assign the Grammarly app to them in your IdP interface.

As the account administrator, you will also be able to log in without SSO by using the following URL:

https://www.grammarly.com/signin?ssoRedirect=false

**Provisioning and Deprovisioning**

**Option 1 - Just-in-time (JIT)**

**Provisioning** - To add new users to your Grammarly Business team, simply assign the Grammarly app to them in your IdP interface. If you assign the app to someone who does not have a Grammarly account, it will be provisioned on their first login.

**Deprovisioning** - To remove users from your Grammarly Business team, go to the Members section of the Grammarly admin panel and click on the trash can icon for the specific user you want to remove.
Option 2 - SCIM

You can configure SCIM (System for Cross-domain Identity Management) for your Grammarly Business account, which automatically provisions / de-provisions members when they are assigned / unassigned the Grammarly application access in your identity provider.

Currently, the identity providers supported for SCIM-based provisioning are Okta, Azure AD, and OneLogin.

For more details and instructions, please refer to our online guides:

- Configuring User Provisioning for your Grammarly Business account
- Configuring Provisioning for Grammarly in Okta
- Configuring Provisioning for Grammarly in Azure AD
- Configuring Provisioning for Grammarly in OneLogin

Notes:

To configure SCIM for your Grammarly Business account, you must already be using SAML Single Sign On.

If you already have Grammarly members in your Grammarly Business account and you turn on provisioning, a reconciliation process needs to happen. Please follow this guide to find additional details about configuring SCIM for your Grammarly Business account.
Grammarly Business for Windows

You can silently install the Grammarly Business for Windows on many computers at once using a Microsoft Intune cloud-based device management tool or Microsoft SCCM.

### What to consider

Minimum requirements - Windows 10 version 1903

Deploying the Grammarly Business for Windows

**Option 1 - Using Intune**

**Pre-requisites:**
- Download the [installer](#)
- Package file into a .intunewin format ([instructions](#))

**MSI Installer**
- x64 perUser [installer](#)
- x86 perUser [installer](#)
- Currently, we do not support perMachine installation

**Guide:**
1) On Azure, go to [Microsoft Endpoint](#) → Apps → Windows Apps and click the **Add** button:
2) Select app type → Windows app (Win32)

3) Add prepared package in .intunewin format:
4) Fill out Application information:

- **Select file**: Grammarly Desktop.intunewin
- **Name**: Grammarly Desktop
- **Description**: Grammarly Desktop application for Windows
- **Publisher**: Grammarly
- **App Version**: Enter the app version
- **Category**:
- **Show this as a featured app in the Company Portal**: Yes
- **Information URL**: Enter a valid url
- **Privacy URL**: Enter a valid url
- **Developer**: 
- **Owner**: 
- **Notes**: 
- **Logo**: Select image

5) Set up a Program installation/uninstallation commands:
Specify the commands to install and uninstall this app:

Install command * ☑
GrammarlyDesktop.exe /S
☑

Uninstall command * ☑
GrammarlyDesktop.exe /S /uninstall
☑

Install behavior ☑
System
User

Device restart behavior ☑
App install may force a device restart

Specify return codes to indicate post-installation behavior:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Code type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Success</td>
</tr>
<tr>
<td>1707</td>
<td>Success</td>
</tr>
<tr>
<td>3010</td>
<td>Soft reboot</td>
</tr>
<tr>
<td>1641</td>
<td>Hard reboot</td>
</tr>
<tr>
<td>1618</td>
<td>Retry</td>
</tr>
</tbody>
</table>

Note: You can use preferred Install behavior - the installation will work in both ways.

6) Select Operation system architecture and Minimum operating system:

☑ App information  ☑ Program  ☑ Requirements  4 Detection rules  5 Dependencies  6 Supers

Specify the requirements that devices must meet before the app is installed:

Operating system architecture * ☑
2 selected

Minimum operating system * ☑
Windows 10 1607

Disk space required (MB)

Physical memory required (MB)

Minimum number of logical processors required

Minimum CPU speed required (MHz)

Configure additional requirement rules

<table>
<thead>
<tr>
<th>Type</th>
<th>Path/Script</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No requirements are specified.

+ Add
7) Setup detection rules to Rules format → Manually configure detection rules → Add

- App information ✔
- Program ✔
- Requirements ✔
- Detection rules ✔
- Dependencies
- Supersede

Configure app specific rules used to detect the presence of the app.

Rules format ✔

<table>
<thead>
<tr>
<th>Type</th>
<th>Path/Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>No rules are specified.</td>
<td></td>
</tr>
</tbody>
</table>

+ Add

Rule type → File
Path → %LOCALAPPDATA%\Grammarly\DesktopIntegrations
File or Folder → Grammarly.Desktop.exe
Detection Method → Registry
Key path → Computer\HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Grammarly Desktop Integrations
Value name → DisplayVersion
Detection method → Version compression
Operator → Greater than or equal to
Value → Use current installer version value
Associated with a 32-bit app on 64-bit clients → Yes
Detection rule

Create a rule that indicates the presence of the app.

Rule type: Registry
Key path: Computer\HKEY_CURRENT_USER\SOFTWARE\Microsoft\...
Value name: DisplayVersion
Detection method: Version comparison
Operator: Greater than or equal to
Value: 0.1.8.28

Associated with a 32-bit app on 64-bit clients: Yes

8) Add Dependencies and Supersedence (preview) if needed
9) Add Assignments:
10) Review and create an app.
Option 2 - Using SCCM

Guide:

1) Copy the GrammarlyInstaller.exe to the folder C:\Sources
2) In the ConfigMgr Console, go to \Software Library\Overview\Application Management\Packages
3) Create a package
4) Important - please specify that this package contains a source files and choose a path to it:
5) Go next and choose Standard program
6) Fill or required fields Next and go further:
**Note:** For Command line please use prefix `/S` for silent installation. For the removal process we need to create a package with command line prefix `/S /uninstall`. 
7) Specify platform requirements:
8) Review the summary and hit Next, after that you should get a successful Completion message:
9) Package is ready to deploy so you can go ahead and install it on required machines:
Grammarly Business for Mac OS

You can silently install the Grammarly Business for Mac on many computers at once using JAMF cloud-based MDM tool (script method).

What to consider
Minimum requirements - macOS 10.12 (Sierra) and up

Deploying the Grammarly Business for Mac

Using JAMF (Script Method)

Guide:

Step 1 - Prepare a script

1) Log in to the Jamf instance and navigate to Computers → Computer Management → Scripts
2) Click the “+New” button in the top right corner.
3) Fill out the Display Name and assign the category in the General tab.
4) Move to Script tab and copy-paste the following script:
#!/bin/sh

currentUser=$(ls -l /dev/console | awk '{ print $3 }')
dmgfile="Grammarly.dmg"
volname="Grammarly"

/bin/echo "--"

# check if logged in user has admin rights
IsUserAdmin=$(id -G $currentUser| grep -v 80)
if [ -n "$IsUserAdmin" ]; then
    grammarly_dir="/Users/$currentUser/Applications"
    /bin/echo "'date': $currentUser is not local admin"
else
    grammarly_dir="/Applications"
    /bin/echo "'date': $currentUser is a local admin"
fi

# check if Applications folder exists
if [ -d "$grammarly_dir" ]; then
    echo "Application folder exists"
else
    echo "Application folder doesn't exist"
    mkdir -R "$grammarly_dir"
    chown -R "$currentUser":staff "${grammarly_dir}"
fi

# check download link and availability of Grammarly CDN
if [ -z "$4" ]; then
    url='https://download-mac.grammarly.com/Grammarly.dmg'
    /bin/echo "No arguments supplied. Using default address"
else
    url=$4
fi

/usr/bin/curl -f -s -I $url &>/dev/null
if [ ![ $? -eq 0 ] ]; then
    /bin/echo "'date': Grammarly Desktop download site ${url} is available."
else
    /bin/echo "'date': Grammarly Desktop download site ${url} is NOT available."
exit 1
fi

/bin/echo "'date': Checking and unmounting existing installer disk image"
/usr/bin/hdiutil detach $(/bin/df | /usr/bin/grep "${volname}" | awk '{print $1} ') -quiet
/bin/sleep 10

/bin/echo "'date': Downloading latest version"
/usr/bin/curl -s -o /tmp/"$dmgfile" "$url"

/bin/echo "'date': Mounting installer disk image"
/usr/bin/hdiutil attach /tmp/"$dmgfile" -nobrowse -quiet

/bin/echo "'date': Installing..."
/bin/echo "${grammarly_dir}/Grammarly Desktop.app"
/usr/bin/sudo -u "$currentUser" cp -R "/Volumes/${volname}/Grammarly Installer.app" "/$grammarly_dir/Grammarly Desktop.app"
/usr/sbin/chown -R "$currentUser":staff "$grammarly_dir/Grammarly Desktop.app"
xattr -r -d com.apple.quarantine "$grammarly_dir/Grammarly Desktop.app"
/bin/sleep 10

/bin/echo "'date': Unmounting installer disk image"
/usr/bin/hdiutil detach $(/bin/df | /usr/bin/grep "${volname}" | awk '{print $1} ') -quiet
/bin/sleep 10

/bin/echo "'date': Deleting disk image"
/bin/rm /tmp/"$dmgfile"

/bin/echo "'date': Setting onboarding deferral"
/usr/bin/sudo -u "$currentUser" defaults write com.grammarly.ProjectLlama deferOnboarding -bool YES

/bin/echo "'date': Running the app"
5) Click the Save button.
Step 2 - Create the Configuration profile to allow Grammarly Desktop app to access the Accessibility

1. Navigate to Computers → Configuration Profiles and click the “+New” button in the top right corner.
2. Fill the General tab:
   - Fill in the Name and Description.
   - Choose a Category according to your organization’s settings.
   - Ensure that the level at which to apply the profile is Computer Level
   - Choose the desired Distribution Method:
     -- Select Install Automatically to push profile to scoped computers immediately;
     -or-
     --Select Make available in the Self Service to let users install it on their own.
3. Navigate to the Privacy Preferences Policy Control and click Configure. Fill out the next:

- **Identifier:** `com.grammarly.ProjectLlama`
- **Identifier Type:** Bundle ID
- **Code Requirement:**
  
  anchor apple generic and identifier "com.grammarly.ProjectLlama" and 
  (certificate leaf[field.1.2.840.113635.100.6.1.9] /* exists */ or certificate 
  [field.1.2.840.113635.100.6.2.6] /* exists */ and certificate 
  leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = 
  W8F64X92K3"

- **Check the “Validate the Static Code requirement” box**
- **Click the “Add” button and select Accessibility from the “App or Service” dropdown**
- **Select Allow from the Access dropdown.**

Navigate to the Scope tab and set the Deployment Targets, Limitation, and Exclusions.
4. Once the Privacy Preferences Policy Control and the Scope are defined, click “Save” to begin the profile enrollment.

**Note:** In order to avoid the Accessibility prompt in the app, the configuration profile should be installed before the application is installed.

**Step 3 - Create and apply the Policy with script**

1. Navigate to Computers → Policies and click the “+New” button in the top right corner.
2. Under the General tab in the New Policy:
   - Enter the policy name;
   - Select the trigger option according to your organization's settings:
     - Check the box near the “Recurring check-in” option to enroll this policy on all scoped computers during the next sync with the Jamf server;
     - or-
     - Check the “Enrollment complete” to trigger the policy immediately after a computer completes the enrollment process;
     - or-
     - Check “Custom” to set up your custom trigger;
     - or-
     - Leave all trigger options unchecked for the installation from the Self Service.
   - Select the Execution Frequency depending on your Jamf organization setting:
     - Select “Once per computer” from the dropdown for initial enrollment
     - or-
     - Select Ongoing for the installation from the Self Service
   - Adjust other fields under the General tab such as Category, Server-side, or Client-side limitations according to your Jamf organization settings.
3. Navigate to Scripts and click Configure
4. Choose the Grammarly Desktop script from the list.
5. [Optional] Adjust Maintenance tab to Update Inventory after the policy installation.
6. Set the Deployment Targets, Limitation, and Exclusions under the Scope Tab.
7. If the policy is set up for the installation from the Self Service, navigate to the Self Service tab:
   - Check the “Make the policy available in Self Service” box;
   - Fill the Self Service Display name, Button Name, Description, and choose Category according to your organization’s settings.
   - You can find the app icon for the Self Service here: Grammarly Icon
Once you complete all the required fields, click the “Save” button to start the policy enrollment.

Browser Extensions

Deploying the Chrome Extension

The Chrome Extension can be deployed in several ways:

**Windows - Option 1 - Group Policy for Chrome Enterprise**

In your Microsoft Windows Group Policy Editor (Computer or User Configuration folder):

1. Go to Administrative Templates → Google → Google Chrome Extensions.
2. Open Extension management settings.
   Tip: If you don’t see the policy, download the latest policy template.
3. Enable the policy and enter its compact JSON data in the text box.
   Note: Enter JSON data as a single line with no line breaks. You can use this third-party JSON compression tool to validate policies and compact them into a single line (example JSON data below).
4. Set the following app and extension policy (sample below).

```
{
    "kbfnbcaeplbcioakkpcpgfkobkghlhen": {
        "installation_mode": "force_installed",
        "update_url": "https://clients2.google.com/service/update2/crx"
    }
}
```

5. After you deploy any Chrome extension policy, check user devices to make sure the policy was applied correctly.
6. On a managed Chrome device, browse to chrome://policy.
7. Click Reload policies.
8. In the top right, in the Filter policies by field box, enter ExtensionSettings.
9. Check the Show policies with no value set box.
10. Under the Chrome policy name next to each extension setting, make sure Status is set to OK.

11. Click Show value and make sure the value field isn't empty.

**Windows - Option 2 - Registry Key**

1. Find or create the following key in the registry:
   - **32-bit Windows:**
     HKEY_LOCAL_MACHINE\Software\Google\Chrome\Extensions
   - **64-bit Windows:**
     HKEY_LOCAL_MACHINE\Software\Wow6432Node\Google\Chrome\Extensions

2. Create a new key (folder) under the Extensions key called "kbfnbcaelbicoolkpcpgfobkghlhen"

3. In your extension key, create a property, "update_url", and set it to the value: "https://clients2.google.com/service/update2/crx"

**Mac - Chrome Policy**

In your Chrome configuration profile:

1. Update the ExtensionSettings key and add the following entry:

   ```xml
   <key>kbfnbcaelbicoolkpcpgfobkghlhen</key>
   <dict>
     <key>installation_mode</key>
     <string>force_installed</string>
     <key>update_url</key>
     <string>https://clients2.google.com/service/update2/crx</string>
   </dict>
   ```

2. After you deploy the extension policy, check user devices to make sure the policy was applied correctly.

3. On a managed Chrome device, browse to chrome://policy.

4. Click Reload policies.

5. In the top right, in the Filter policies by field box, enter ExtensionSettings.

6. Check the Show policies with no value set box.

7. Under the Chrome policy name next to each extension setting, make sure Status is set to OK.
8. Click Show value and make sure the value field isn't empty.
9. Launch the browser and go to chrome://extensions; you should see the extension listed.

Deploying the MS Edge Extension

Option 1 - Using a Group Policy (Windows)

Prerequisite: Download the Edge Administrative Templates from:


You will need to fill out the required fields and click the "GET POLICY FILES" button

1. Open Group Policy Management
2. Create a new GPO for the policy (or add to an existing policy)
3. Right-click your GPO, and select "Edit"
4. You will then see the “Group policy management Editor” window
5. Expand the Computer/User configuration tree on the left-hand side, depending on how you wish to configure your policy
6. Right-click on “Administrative templates” and select “Add/Remove Templates”
7. Click on “Add”
8. Browse to the templates you downloaded in step 2 and open the “msedge.adm” template that is relevant to your operating system and Language.
9. Once uploaded, expand to the following path (may vary depending on your OS) on the left of the Group policy management editor: “Computer/User configuration > Policies > Administrative templates > Classic administrative templates > Microsoft Edge > Extensions”

10. On the right-hand side, select “Configure the list of force-installed extensions”
11. Right-click, and select “Edit”
12. Mark the “Enabled” button
13. Click the “Show…” button
14. Under the “Value” column, enter the following:
   `kbfnbcaeplbcioakkkpcpgfkbkghlhen;https://clients2.google.com/service/update2/crx`
   (This is the Google Chrome Based extension which will also work on MS Edge v77+)
   
   **Note:** Make sure you have no blank spaces in the Value Box Below

15. Click “OK”, followed by “Apply”
16. Back in the Group Policy Management window, assign this policy to users/computers as normal
17. Users affected by this GPO should now see the Edge extensions installed and enabled automatically (once the GPO has updated on their machine)

**Option 2 - Registry Key (Windows)**

1. Find or create the following key in the registry:
   - **32-bit Windows:**
     HKEY_LOCAL_MACHINE\Software\Microsoft\Edge\Extensions
   - **64-bit Windows:**
     HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Edge\Extensions

2. Create a new key (folder) under the Extensions key called “cnlefmmeadmmeatmgahbnfmdpbfbd”

3. In your extension key, create a property, "update_url", and set it to the value:
   "https://edge.microsoft.com/extensionwebstorebase/v1/crx"
Managed Mode

What is Managed Mode?

Managed Mode ensures that only users that belong to your Grammarly Business account are able to sign in to the Grammarly Browser Extensions, or to the MS Office Add-In (Windows). When this setting is enabled company employees won’t be able to use their personal Grammarly accounts at work through those clients.

Currently, this is only available for: (1) browser extensions for Google Chrome, Mozilla Firefox, and Microsoft Edge, (2) Grammarly MS Office Add-In (Windows).

Managed Mode will also deactivate the clients and stop sending text to our servers if:

- The user is not logged in, or
- If the user is logged in but is not part of the relevant organization

Note: Users will still be able to log in with personal accounts through our sign-in page, however they won't be able to use the clients.
Enabling Managed Mode

**Prerequisite**: SSO needs to be enabled for the organization. You will also need a **Grammarly Enrollment Token** that you can obtain from your account manager or the Grammarly Support team.

Chrome Extension (Windows)

The following key needs to be configured in the Windows Registry:

- \[HKEY_LOCAL_MACHINE\Software\Policies\Google\Chrome\3rdparty\extensions\kbfnbcaeplbcioakkpcpgf Kobkghlhen\policy\]

The key should include the following value:

- "GrammarlyEnrollmentToken"="{UUID}"

Note: **kbfnbcaeplbcioakkpcpgf kobkghlhen** is the ID of the official Grammarly Chrome Extension and the {UUID} should be replaced with your Grammarly Enrollment Token.

The Grammarly Chrome Extension will switch to Managed Mode when it discovers a valid enrollment token and obtains the settings for your organization from our servers.

Once this is configured, here is what should be changed compared to the extension's default behavior:
1. Instead of a regular “Sign in” button, it will show a “Sign in with SSO” button that leads directly to the IdP URL.

2. After login, it will display the “Managed by {Your company name} ” stripe in the bottom of the extension’s popup dialog (the one that you see when clicking the G icon in the extensions Chrome bar). Hovering over the company name would also display the enrollment token that it read from the settings.
Note: Make sure to restart Chrome after modifying the registry so this change can be reflected. For Windows 11, you will need to reboot or reload policies by accessing chrome://policy/ in Chrome browser.

Chrome Extension (Mac)

This can be done by creating a plist file on the /Library/Managed Preferences/{username} folder.

File name:
com.google.Chrome.extensions.kbfnbcaeplbcioakkpcpgfkobkghlhen.plist

File contents:
Note: Make sure to replace the {UUID} with your Grammarly Enrollment Token.

Finally, make sure to convert the XML file to binary using the following terminal command:

```
plutil -convert binary1 {filename} com.google.Chrome.extensions.kbfnbcaepbcioakkpcpgfkobkghlhen
```

If using JAMF or a similar tool, you can deploy a custom configuration profile (instructions) with a the following Preference domain:

```
com.google.Chrome.extensions.kbfnbcaepbcioakkpcpgfkobkghlhen
```

You can then upload a plist file with the file content above in gray. The file should be in XML format.

Edge Extension (Windows)

The following key needs to be configured in the Windows Registry:

- [HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Edge\3rdparty\extensions\cnlefmmeadmemmdciolhbnfeacpdfbkd\policy]
The key should include the following value:

- "GrammarlyEnrollmentToken"="{UUID}"

Note: `cnlefmmeadmemmdciolhbnfeacpdfbkd` is the ID of the official Grammarly Edge Extension and the `{UUID}` should be replaced with your Grammarly Enrollment Token.

The Grammarly Edge Extension will switch to Managed Mode when it discovers a valid enrollment token and obtains the settings for your organization from our servers.

Note: Make sure to restart the Edge browser after modifying the registry so this change can be reflected.

**Edge Extension (Mac)**

This can be done by creating a plist file on the `/Library/Managed Preferences/{username}` folder.

File name:
com.microsoft.Edge.extensions.cnlefmmeadmemmdciolhbnfeacpdfbkd.plist

File contents:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
.plist version="1.0">
<dict>
  <key>GrammarlyEnrollmentToken</key>
  <string>{UUID}</string>
</dict>
</plist>
```
Note: Make sure to replace the {UUID} with your Grammarly Enrollment Token.

The last step is to deploy your plist to your users' Mac devices using your preferred MDM provider, such as Microsoft Intune. For instructions see Deploy your plist.

For Jamf upload the .plist file as a Custom Settings payload.

Firefox Extension (Windows)

This can be done by creating a JSON file following the Firefox Managed storage manifests instructions.

- Under the data section add the
  "GrammarlyEnrollmentToken"="{UUID}"

where the {UUID} is your Grammarly Enrollment Token.

- Under the name section add the Firefox extension ID:
  87677a2c52b84ad3a151a4a72f5bd3c4@jetpack

File content example:

```json
{
  "name": "87677a2c52b84ad3a151a4a72f5bd3c4@jetpack",
  "description": "ignored",
  "type": "storage",
  "data":
    {
      "GrammarlyEnrollmentToken": "{UUID}"}
}
```
Save and put this file in any location on the computer.

Configure the path to the file in the registry:

[HKEY_CURRENT_USER\SOFTWARE\Mozilla\ManagedStorage\{name}]

where the {name} equals the name value from the file.

Note: Make sure to restart the Firefox browser after modifying the registry so this change can be reflected.

Firefox Extension (Mac)

This can be done by creating a JSON file following the Firefox Managed storage manifests instructions.

- Under the data section add the
  "GrammarlyEnrollmentToken"="{UUID}"

where the {UUID} is your Grammarly Enrollment Token.

- Under the name section add the Firefox extension ID:
  87677a2c52b84ad3a151a4a72f5bd3c4@jetpack

File content example:

```json
{
  "name": "87677a2c52b84ad3a151a4a72f5bd3c4@jetpack",
  "description": "ignored",
  "type": "storage",
  "data":
  {
    "GrammarlyEnrollmentToken": "{UUID}"
  }
}
```
For global visibility, save and put this file in:

/Libary/Application Support/Mozilla/ManagedStorage/<name>.json

For per-user visibility, save and put this file in:

~/Library/Application Support/Mozilla/ManagedStorage/<name>.json

Note: Make sure to restart the Firefox browser after modifying the registry so this change can be reflected.