Practical Guide: Automating Dependency Updates with MintMaker and Renovate

Introduction

Keeping dependencies up-to-date is a critical part of maintaining a secure and healthy codebase. However, doing this manually is tedious and often gets deprioritized. To solve this, we are adopting **MintMaker** (powered by **Renovate**) across our repositories.

This guide explains how the setup works, breaks down our configuration, and provides instructions on how to adopt it for your own repositories.

What is MintMaker & Renovate?

In short:

- **Renovate** is the engine. It scans your repository (e.g., `package.json` for React/TypeScript projects), detects outdated dependencies, and creates Pull Requests (PRs) with updates.
- **MintMaker** is the service (internal to our CI/CD ecosystem) that triggers and manages Renovate runs for us.

Instead of us manually checking npm for new versions, the bot does it automatically.

Configuration Breakdown (`renovate.json`)

To enable this, we add a `renovate.json` file to the root of the repository. Below is the standard configuration we are rolling out, with an explanation of each part.

```
{
  "$schema": "https://docs.renovatebot.com/renovate-schema.json",
  "extends": ["config:recommended"],
  "baseBranches": ["main"],
  "tekton": {
    "schedule": ["at any time"]
  },
  "packageRules": [
    {
      "matchManagers": ["npm"],
      "rangeStrategy": "bump"
    },
    {
      "matchManagers": ["npm"],
      "matchUpdateTypes": ["minor", "patch"],
      "groupName": "npm minor and patch dependencies",
      "enabled": true
    },
    {
      "matchManagers": ["npm"],
      "matchUpdateTypes": ["major"],
      "enabled": false
    }
  ]
}
```

Key Settings Explained

1. **`extends` : `["config:recommended"]`**
   - We don't start from scratch. We inherit a base set of best practices from the Renovate team (e.g., enabling the "Dependency Dashboard" issue, semantic commit prefixes, etc.).
2. **`baseBranches` : `["main"]`**
   - Ensures the bot only targets our production branch.
3. **`tekton` : `{"schedule": ["at any time"]}`**
   - This is specific to Tekton pipeline definitions. It tells the bot that updates related to Tekton tasks/pipelines can be processed immediately, bypassing default scheduling restrictions if any exist.
4. **`packageRules` (The Logic)**
   - **Rule 1 (`rangeStrategy` : "bump")**: By default, Renovate might just update the lockfile (`package-lock.json`) if a version satisfies the existing range (e.g., `^1.0.0`). Setting this to `"bump"` forces it to also update the version explicitly in `package.json` . This keeps our manifests declarative and up-to-date.

- **Rule 2 (Grouping Minor/Patch)**: This is a huge noise reduction feature. Instead of getting 10 separate PRs for 10 different libraries upgrading from `1.0.1` to `1.0.2`, Renovate will group them into a single PR named "npm minor and patch dependencies".
- **Rule 3 (Major Updates Disabled)**: Major updates (e.g., React 17 -> 18) often contain breaking changes. We have explicitly set `"enabled": false` for these. This means the bot won't spam us with potentially breaking PRs automatically. We prefer to handle major upgrades manually when we have the capacity.

Practical Tips for the Team

- **Reviewing PRs**: You will see PRs coming from the bot user. Treat them like normal code changes. Check the **Release Notes** included in the PR description (Renovate is great at fetching changelogs).
- **React & TypeScript**: Since we use npm, this config works out of the box. Renovate understands `devDependencies` vs `dependencies` automatically.
- **Dependency Dashboard**: Renovate will create a "pinned" issue in the repository called the *Dependency Dashboard*. You can use this issue to see pending updates, force a retry, or manually request a major update that is otherwise disabled.

How to Implement

1. Copy the `renovate.json` from the example above.
2. Create a PR adding this file to the root of your repository.
3. Once merged, MintMaker will pick it up on its next cycle (usually within an hour) and you should see an "Onboarding" PR or the Dependency Dashboard issue appear.

Resources

For more details on configuration options and advanced features, please refer to the official documentation:

- Renovate Documentation - Main source for configuration options and behavior.