

**the agile
monkeys.**

The Agile Monkeys

Backend Challenge with Django

API Test- The CRM Service

The Challenge:

Welcome to The Agile Monkeys hiring process! We're excited to have you here.

For this phase, we require you to build a **synthetic API project using Django**.

Your task is to **create a REST API** to manage customer data for a small shop. This API will act as the backend for a CRM interface being developed by another team. As the lead developer on the backend, you'll be responsible for the design and implementation of the API. Here's what we're looking for:

- The API should be only accessible by a registered user by providing an authentication mechanism.
- A user can:
 - List all customers in the database.
 - Get full customer information, including a photo URL.
 - Create a new customer:
 - A customer should have at least a name, surname, ID, and photo field.
 - Name, surname, and id are required fields.
 - Image uploads should be able to be managed.
 - The customer should have a reference to the user who created it.
 - Update an existing customer:
 - The customer should hold a reference to the last user who modified it.
 - Delete an existing customer.
- An admin can also:
 - Manage users:
 - Create users.
 - Delete users.
 - Update users.
 - List users.
 - Change admin status.

While the design is totally up to you, we recommend sticking to the tools and libraries you're most comfortable with.

We'll invite you to a Slack channel where you can communicate with our team. Feel free to ask any questions and explain the choices you make during the process!

Once you're ready, please create a Git repository for the project on GitHub or Bitbucket and share the link with the team in the Slack channel.

What we will evaluate in this test:

- **Code Quality:** We want to see clean, readable, and simple code. Focus on good naming, structure, and sticking to framework standards. Write it like it's going live in production.
- **Software Architecture:** Aim for low coupling and easy-to-change code. Use design patterns and best practices where they make sense, and show good use of the framework or language.
- **Team Communication:** Keep us in the loop! Clear communication and asking questions when needed are key.
- **Basic Security Practices:** Make sure you've got the basics covered, like authentication, authorization, and protection against SQL injection and XSS.
- **Completeness and Accuracy:** The solution should meet all the requirements and work as expected without any major issues.

Optional features: If you'd like to go the extra mile, here are some optional features you can include. These are not required, but would definitely show off some additional skills:

- A good **README** file with a getting started guide.
- **Tests** implemented for your solution.
- Making the **project set-up** easier for newcomers (streamlining the process).
- Following the **Twelve-Factor App principles** (<https://12factor.net>) to ensure scalability.
- Using **OAuth 2 protocol** for authentication (you can use a third-party public OAuth provider).
- Making the project **Continuous Deployment** ready using a provider like AWS.
- Using **Docker, Vagrant, or other tools** to simplify the development environment configuration.

We're excited to see your approach to this challenge—good luck, and have fun! If you need anything along the way, don't hesitate to reach out!

The Agile Monkeys.