

Red Technologies Coding Challenge

Thank you for your interest in working for Red Technologies. As a rapidly growing company, we are excited to speak with and learn from developers who are as passionate about programming as we are. As you traverse the challenge, put your best foot forward and feel free to comment or ask questions. If an objective is within your reach but can't be implemented in the time given, feel free to walk us through how you would approach it conceptually.

The challenge is broken into three scoring tiers. The first is **Mandatory** objectives. These are items we consider day-to-day development tasks that may present some challenge but ideally are achievable in the time given. The second is **Above Average**, implying there will be no penalty for failure to achieve, but would help a candidate stand out against competition. The final is **Exceptional**. This is for candidates with a wide variety of experience who want to present a fully featured, hosted application.

BACKEND

■ Project Overview

- Using .NET/.NET Core (3.1 or higher [5, 6, etc.]) and C#, create an "Orders" web API to view, create, update, and delete orders
 - An additional endpoint will be created to search by "Customer"

■ Deliverables

- Implement an external data store for the orders data
 - Files, normalized/non-normalized databases, caching, etc., are acceptable
 - Preference will be given to normalized database implementations (SQL scripts are an ok deliverable)
- Implement 5 endpoints
 - Create, read, update and delete endpoints for orders
 - Endpoints should offer basic input validation and error handling
 - A search endpoint to filter by Order Types
 - Order Types are Standard, SaleOrder, PurchaseOrder, TransferOrder, and ReturnOrder
- Create and implement at least a repository layer using Dependency Injection (DI)
 - Feel free to use .NET's native features or a 3rd party package if preferred
- Entity models should *not* propagate to the end user, so mapping between entities and user-friendly models should occur



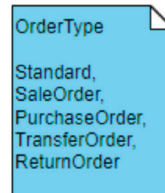
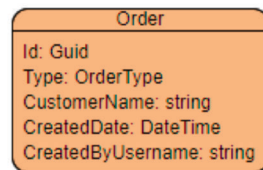
If you have any questions, please contact Alex Johnson
ajohnson@redtms.com



Red Technologies Coding Challenge

BACKEND (CONTINUED)

- Here is the layout of the Order entity



*Note a table for OrderType is not required

■ Mandatory Features

- A fully functional set of endpoints to match the expected deliverables
- Any kind of external data store for source data
- Basic DI Implementation for a data ingress class

■ Above Average Features

- A normalized data store for application entities. An SQL script is an acceptable deliverable. Special consideration is given for knowledge/implementation of Entity Framework.
 - Code-first or Database-first is acceptable
- Install Swagger on top of the project
- Demonstrated knowledge of DI lifetime management
 - A transient or scoped service layer to handle business logic or repository passthrough is acceptable
- Unit tests for Controllers and DI classes

■ Exceptional Features

- All of the Above Average deliverables
- Host your Database and API in a location accessible remotely (i.e. publicly)
- Implement authentication (username/password, token, etc.) for the API
- Utilize containerization and/or a deployment pipeline to host the app programmatically



If you have any questions, please contact Alex Johnson
ajohnson@redtms.com



Red Technologies Coding Challenge

FRONTEND

■ Resources

- A web API is available for your use to list, create, update, search, and delete example orders for your project app
- The URL is <https://red-candidate-web.azurewebsites.net/index.html> which will take you to a Swagger page describing the API
- The Swagger page is read only (no authorization for API calls)
- To make requests use an HTTP header with the key 'ApiKey' and a value of 'b7b77702-b4ec-4960-b3f7-7d40e44cf5f4'
- **NOTE:** If applying/attempting as a fullstack candidate, your frontend challenge is expected to use a working implementation of the backend challenge

■ Deliverables

- Create a view of order entities in-memory on the page
 - The finished product should resemble the mockup on the next page
- Implement a feature where users can create new order entities
- Implement a simple dropdown filter to limit the order view to certain order types
- Implement a simple textbox search to isolate an order by ID
- Implement a delete feature by which a user can remove one or many orders at once



If you have any questions, please contact Alex Johnson
ajohnson@redtms.com



Red Technologies Coding Challenge

FRONTEND (CONTINUED)

■ Mandatory Features

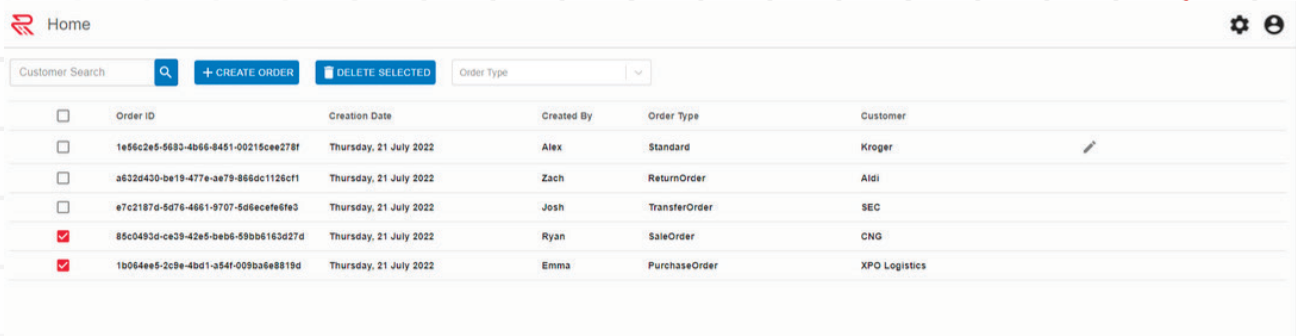
- Fully functional web page/application which behaves as described on the previous page
- Usage of a component library (Material UI recommended)
- Reasonably responsive behavior for different screen widths

■ Above Average Features

- Implement frontend testing using a framework of your choice
- Implement a command to build/run the application with a console command
- Implement Redux (or other state management) and use it to create a 'save draft' feature for order creation
 - It's acceptable if drafts don't survive page refreshes

■ Exceptional Features

- All the Above Average deliverables
- Host the app in a publicly-available location
- Implement containerization and/or pipelines to automate the app deployment
- Design a custom view/theme for the app including animations or other user friendly features



If you have any
questions, please
contact Alex Johnson
ajohnson@redtms.com

