**ReachLocal Japan Programming Test Instructions**

Instead of whiteboard coding or algorithm quizzes, we would like to see how you work with a small project that uses fundamentals similar to what we handle daily. While going beyond this spec is great, the purpose of this is primarily to see how you address these specific requirements. We place a high value on maintainability and idiomatic use of programming languages as well as the UI designing skills; please see the second page for general expectations.

This project should take between 3 to 4 hours.

1. Create a single page application
2. Consume data from https://financialmodelingprep.com/api/v3/historical-price-full/AAPL,GOOGL,AMZN
3. Create component with comparison chart between companies close stock price with ability for user to filter data by date and company
4. Create component with buying stock form for specified company with total price calculation based on latest close price and  input volume.
    a. Validate that user can buy not more than 1000 stocks or total price not more than 1 000 000$
    b. Validate that user can buy stock only when stock market is open

       ( Tokyo Stock Exchange, Monday to Friday, 09:00-11:30, 12:30-15:00 )

- Create component which will display a list of user order cart with ability to edit added items

5. Either make your project public on **BitBucket** and provide the link or share the private repository with us (username is *zhelinrlj*)

While this obviously needs to be your code, don’t hesitate to use libraries and code snippets where appropriate. Just make sure you leave a comment with the source if you choose to use snippets, and remember that the point of this exercise is for us to see your own work.

During the technical interview, we’ll ask you to add a minor feature to this project so we can watch you work. The feature will be picked based on your particular project.

**Expectations for Javascript and other technique usage**

- Code should be written using React and ES6 (Please, push both source code and compiled files )
- Utilize Redux to exchange data between components
- Feel free to use any additional tools and libraries
- Include list of installed packages, bundler configuration file
- Write a few tests for your code
- Include documentation for project
- Gracefully handle errors, external data and invalid user input