

## Coding Challenge Overview:

Each of our Paylocity product teams operates like a small startup, empowered to deliver business value in whatever way they see fit. Because our teams are close knit and fast moving it is imperative that you are able to work collaboratively with your fellow developers.

This coding challenge is designed to allow you to demonstrate your abilities and discuss your approach to design and implementation with your potential colleagues. You are not expected to spend more than a few hours on this project, and you are free to use whatever technologies you prefer but please be prepared to discuss the choices you've made. **The most important part of this challenge is to use your work as a jumping off point for a broad and deep conversation with our developers.**

We are expecting candidates to typically spend about 2-3 hours on this exercise, and we realize that this may lead to an incomplete implementation. Please take this as an opportunity to demonstrate the best of your abilities – feel free to mock out or skip pieces of the implementation and focus your time on exactly what you'd like to show us. We're not looking for you to give up your weekend for this project, we want something to start a meaningful conversation.

## Business Need:

One of the critical functions that we provide for our clients is the ability to pay for their employees' benefits packages. A portion of these costs are deducted from their paycheck, and we handle that deduction. Please demonstrate how you would code the following scenario:

- The cost of benefits is \$1000/year for each employee
- Each dependent (children and possibly spouses) incurs a cost of \$500/year
- Anyone whose name starts with 'A' gets a 10% discount, employee or dependent

Because this calculation will be needed in multiple places (multiple web applications, mobile devices, etc...) we will need a well-designed domain API to serve this data in a scalable fashion. Employees may use one or more devices to choose their benefits package as part of a multi-step process that involves inputting dependents and need a preview of the costs, and administrators need to preview payroll before it is run to double check the numbers. The costs may change in between, so the calculation needs to reflect the current state of the calculation, rather than the state at the time the employee entered it.

This is of course a contrived example. We want to know how you would design the API and backing data store, and then implement the class structure and calculations.

You can make the following assumptions:

- All employees are paid \$2000 per paycheck before deductions
- There are 26 paychecks in a year