Groovy for EPM Planning Cloud / EPBCS / PBCS Training

Before I head out to California to do another Groovy workshop, and to celebrate the completion of all the modules, I wanted to celebrate by offering a discount. This week only, get a $150 discount. I have not talked about this in an official post, but it has gotten a ton of great feedback. The discount will be available through Friday. It is time to get excited! I am adding a case study with full code to make your application a real-time what-if application with full calculations and undo operations before the data is committed.

Prospective Audience

The industry is made up of a wide range of backgrounds. We are surrounded by financial resources with an aptitude toward technology. We have IT trained resources that have been exposed to coding. And we have engineers with a strong java background. All of these resources are like minded in that we learn by doing and like to find the answers ourselves. Many will get an example from an industry peer and replicate that example. Sometimes the example isn’t efficient, or it gets used for purposes it was not intended to be used. The java experts might stick with what they know and write groovy as if it was java.

The result is perpetuating scripts that are less than optimal in the community, building scripts that are not flexible causing issues when data forms change, and code that can’t easily be reused. Both of these strategies will get the job done but neither is the good approach.

So, why take Groovy for EPM Cloud Planning when you think you have a strong background in groovy calculations?

- So you don’t write a script like this to get the members in the point of view [pause] when it can be written like this.
- So you don’t build a dynamic fix statement like this [pause] when it can be written like more efficiently.
- So you dynamically reference all run time prompts that can be reused in all your scripts [pause] rather than hard code the references that have to be changed for every rule.

So you know that there are an infinite amount of things that can be done rather than being limited by only what you have seen. When you go through Groovy for EPM Cloud Planning, you will be able to do things like

- Update outlines sourced from a rest source in a calculation.
- Allow power users to limit what can be entered unless the planner documents a reason for the override in a cell comment
- You can even build a what-if process that uses existing architecture and allows results to be viewed and stored virtually before the data is committed to the database with full undo or cancel abiliy
- Or, email administrators immediately when a user executed calculation that doesn’t finish successfully

This is just a taste!

Why Is This Different

Other internal initiatives and in class or web learning covers the basics of the Planning API. Understanding the components of Groovy to effectively use the Planning API is missed. Without that knowledge, implementation specialists will never

be as effective as they should be. Taking the approach of bypassing learning the Groovy basics isn’t is like learning advanced algebra without knowing how to add and subtract. You will go through the motions, maybe get the right answer, but you will never understand why you are doing what you are doing and your path to the solution will be long and frustrating. You won’t be able advance your knowledge without seeing examples from somebody else and your creative solutions will be minimized to replicating what others have done.

Course Content

This class is designed to be self-paced and will give participants everything they need to know about Groovy to build complex solutions. Everyone will leave with the knowledge of reading and understanding the Java Docs, enabling them to expand their abilities as enhancements are released. The following topics are covered.

- Understanding Groovy variables
- Understanding Groovy syntax and scripting basics
- Understanding Groovy error trapping and logging
- Optimizing Essbase calculations by dynamically creating scripts based on edited data
- Optimizing data maps and smart pushes
- Interacting with run-time prompts to validate user input
- Interacting with data forms to validate user input
- Building calculations for ASO play types
- Creating real time play type data synchronization
- Creating real time consolidated application reporting

Module Format

Every module is organized in the same way, so those attending have a consistent and positive experience. They start with an introduction and a review of the module objectives. Every module includes instruction that covers both basic and

advanced content in a way that is easy to understand for both the novice and the experienced. The modules include a narrated presentation of the material. The concepts are demonstrated with an application modeled after Best Buy* products, departments, and stores, and I apply the teachings to real-world scenarios. Each module concludes with reiterating the module objectives and gives the attendees exercises to complete to reinforce what was learned. Every module is supported by downloadable copies of every line of code that was discussed, as well as the answers to the learning labs when applicable.

Real-World Learning

When participants complete the modules, they will not only be capable of building end-to-end solutions with Groovy, they will also leave with critical information I have consolidated by using Groovy since its release. There are challenges I share that have taken me months to overcome. These include roadblocks developers encounter, from data precision issues using data grids to optimization techniques when calculating ASO play types.

That’s A Wrap

If you have any questions, you are more than happy to reach out to me directory, post a question, or email me. Have a great week!