

JodaTime to java.time.* Migration Plan

Introduction

This document outlines the migration plan for converting JodaTime usage to java.time.* in the MagicBane Server codebase. The ArcMineWindowAvailableTimeMsg.java file has been successfully migrated as a test case, and serves as the reference for the migration approach to be used across all other JodaTime references.

Migration Details

The migration from JodaTime to java.time.* involves converting specific classes and methods:

JodaTime Class	java.time.* Equivalent	Notes
DateTime	Instant	Represents a point in time
Period	Duration	Represents a duration between two points
Seconds	Duration	Represents a duration in seconds

Changes Made in ArcMineWindowAvailableTimeMsg.java

The migration was successfully completed for ArcMineWindowAvailableTimeMsg.java with the following changes:

1. Replaced import 'org.joda.time.DateTime' with 'java.time.Instant'
2. Replaced import 'org.joda.time.Duration' with 'java.time.Duration'
3. Replaced import 'java.time.temporal.ChronoField' with 'java.time.temporal.ChronoField'
4. Changed field type from DateTime to Instant
5. Changed field type from Period to Duration
6. Replaced DateTime.now() with Instant.now()
7. Replaced Period.between() with Duration.between()
8. Replaced Period.from() with Duration.ofDays() or similar java.time methods
9. Updated method calls and calculations to use java.time.* APIs

Migration Strategy

For the complete migration across all 30 files using JodaTime, the following approach should be taken:

1. Create a new branch 'ai-refactor-jodatimeFull' for tracking the migration
2. Use ArcMineWindowAvailableTimeMsg.java as a reference template for migration patterns
3. For each file with JodaTime imports, perform the following conversions:
 - Replace import statements from org.joda.time.* to java.time.*
 - Replace DateTime with Instant (when representing a point in time)
 - Replace Period with Duration (when representing a duration)

- Replace Seconds with Duration (when representing a duration)
- Update method calls and calculations accordingly
- 4. Test each migration thoroughly to ensure functionality remains intact
- 5. Update the JodaTime_References_Tracking.txt file as files are completed

Test Plan

To ensure the migration works correctly in-game:

1. Test the ArcMineWindowAvailableTimeMsg functionality specifically
2. Run comprehensive unit tests for all migrated files
3. Perform integration testing with related systems (guild management, building systems)
4. Test in a local game environment to verify proper behavior
5. Validate that time-based calculations still work correctly

File Tracking

The migration progress is tracked using JodaTime_References_Tracking.txt file. This file contains all 30 files that need to be migrated and their current status.

Note: The JodaTime_References_Tracking.txt file should not be committed to version control, as it's intended purely for team reference during the migration process.