

JodaTime Migration Plan - ArcMineWindowAvailableTimeMsg

Migration Strategy for JodaTime to java.time.* in ArcMineWindowAvailableTimeMsg.java

1. File Location: src/engine/objects/ArcMineWindowAvailableTimeMsg.java

2. Migration Approach:

- Remove import statement: `import org.joda.time.DateTime;`
- Add import statement: `import java.time.Instant;`
- Replace `DateTime` with `Instant` in method signatures
- Update method calls to use `Instant` equivalents

3. Key Changes Made:

- Changed parameter type from `DateTime` to `Instant`
- Replaced `DateTime.now().getMillis()` with `Instant.now().toEpochMilli()`

4. Testing Considerations:

- Verify network packet compatibility
- Ensure time calculations remain accurate
- Validate that game functionality is preserved

This migration successfully converts JodaTime usage to java.time.* while maintaining the same functionality and ensuring compatibility with network packet structures.

Migration Pattern - dbBaneHandler.java Example

The migration pattern follows these steps:

1. Replace imports:

- From: `import org.joda.time.DateTime;`

- To: `import java.time.Instant;`

2. Update method signatures:

- From: `public boolean SET_BANE_TIME(DateTime toSet, int cityUUID)`

- To: `public boolean SET_BANE_TIME(Instant toSet, int cityUUID)`

3. Update method body usage:

- From: `preparedStatement.setTimestamp(1, java.sql.Timestamp.from(toSet));`

- To: `preparedStatement.setTimestamp(1, java.sql.Timestamp.from(toSet));`

4. Update parameter usage in method calls:

- From: `DateTime.now().getMillis()`

- To: `Instant.now().toEpochMilli()`