

Project:

Oscar-Johnson-IndieProject

Scope:

Project 'Oscar-Johnson-IndieProject'

Profile:

Default

Results:

Enabled coding rules: 459

- FindBugs: 346
- Checkstyle: 30
- PMD: 83

Problems found: 279

- FindBugs: 0
- Checkstyle: 241
- PMD: 38

Time statistics:

- Started: Sat Apr 27 22:23:31 CDT 2024
 - Initialization: 00:00:00.008
 - Checkstyle: 00:00:00.190
 - FindBugs: 00:00:02.554
 - PMD: 00:00:00.497
 - Gathering results: 00:00:00.079
- Finished: Sat Apr 27 22:23:34 CDT 2024

Detailed Results:

Oscar-Johnson-IndieProject

- Efficiency
 - Hide Utility Class Constructor
 - CognitoJWTParser
 - Utility classes should not have a public or default constructor. - line: 17
 - SessionFactoryProvider
 - Utility classes should not have a public or default constructor. - line: 15
 - Unnecessary Local Before Return
 - Auth
 - Consider simply returning the value vs storing it in local variable 'request' - line: 258
- Maintainability
 - Avoid Duplicate Literals
 - Auth
 - The String literal "/errorPage" appears 4 times in this file; the first occurrence is on

line 133 - line: 132

- CognitoJWTParser
 - The String literal "\" appears 4 times in this file; the first occurrence is on line 34 - line: 33
- User
 - The String literal "user" appears 4 times in this file; the first occurrence is on line 48 - line: 47
- UsersHTML
 - The String literal "" appears 12 times in this file; the first occurrence is on line 43 - line: 42
 - The String literal "" appears 12 times in this file; the first occurrence is on line 43 - line: 42
- Cyclomatic Complexity
 - AccountInsertUpdate
 - Cyclomatic Complexity is 11 (max allowed is 10). - line: 33
- Loose coupling
 - ApiNinjas
 - Avoid using implementation types like 'ArrayList'; use the interface instead - line: 39
 - ExerciseDb
 - Avoid using implementation types like 'ArrayList'; use the interface instead - line: 30
 - FetchApiResponse
 - Avoid using implementation types like 'ArrayList'; use the interface instead - line: 24
- Preserve Stack Trace
 - Auth
 - New exception is thrown in catch block, original stack trace may be lost - line: 198
 - New exception is thrown in catch block, original stack trace may be lost - line: 201
 - CognitoJWTParser
 - New exception is thrown in catch block, original stack trace may be lost - line: 37
 - New exception is thrown in catch block, original stack trace may be lost - line: 39
 - New exception is thrown in catch block, original stack trace may be lost - line: 58
 - New exception is thrown in catch block, original stack trace may be lost - line: 60
 - New exception is thrown in catch block, original stack trace may be lost - line: 77
 - New exception is thrown in catch block, original stack trace may be lost - line: 98
- Unused Imports
 - Auth.java
 - Unused import - java.time.LocalDate. - line: 36
 - CreateExercisesServlet.java
 - Unused import - org.apache.logging.log4j.LogManager. - line: 2
 - CreateExercisesServlet.java
 - Unused import - org.apache.logging.log4j.Logger. - line: 3
 - CreateExercisesServlet.java
 - Unused import - java.util.List. - line: 13

- DeleteRecord.java
 - Unused import - java.util.Date. - line: 12
- ExerciseDbServlet.java
 - Unused import - fit.app.pojo.ApiNinjaResult. - line: 3
- ExerciseDbServlet.java
 - Unused import - org.apache.logging.log4j.LogManager. - line: 5
- ExerciseDbServlet.java
 - Unused import - org.apache.logging.log4j.Logger. - line: 6
- ExportUserServlet.java
 - Unused import - fit.app.database.GenericDao. - line: 5
- InitialPropertiesServlet.java
 - Unused import - org.apache.logging.log4j.LogManager. - line: 3
- InitialPropertiesServlet.java
 - Unused import - org.apache.logging.log4j.Logger. - line: 4
- RecentRecordLoader.java
 - Unused import - java.util.ArrayList. - line: 5
- Unused local variable
 - Auth
 - Avoid unused local variables such as 'keyId'. - line: 185
 - Avoid unused local variables such as 'alg'. - line: 186
 - GenericDao
 - Avoid unused local variables such as 'root'. - line: 102
- Unused Private Field
 - UsersHTML
 - Avoid unused private fields such as 'logger'. - line: 15
 - UsersText
 - Avoid unused private fields such as 'logger'. - line: 15
- Visibility Modifier
 - Auth
 - Variable 'properties' must be private and have accessor methods. - line: 57
 - Variable 'CLIENT_ID' must be private and have accessor methods. - line: 62
 - Variable 'CLIENT_SECRET' must be private and have accessor methods. - line: 67
 - Variable 'OAUTH_URL' must be private and have accessor methods. - line: 72
 - Variable 'LOGIN_URL' must be private and have accessor methods. - line: 77
 - Variable 'REDIRECT_URL' must be private and have accessor methods. - line: 82
 - Variable 'REGION' must be private and have accessor methods. - line: 87
 - Variable 'POOL_ID' must be private and have accessor methods. - line: 92
 - Variable 'jwks' must be private and have accessor methods. - line: 97
 - GenericDao
 - Variable 'sessionFactory' must be private and have accessor methods. - line: 25
 - LogIn
 - Variable 'properties' must be private and have accessor methods. - line: 26
 - Variable 'CLIENT_ID' must be private and have accessor methods. - line: 31
 - Variable 'LOGIN_URL' must be private and have accessor methods. - line: 36

- Variable 'REDIRECT_URL' must be private and have accessor methods. - line: 41
 - SignUpServlet
 - Variable 'properties' must be private and have accessor methods. - line: 26
 - Variable 'CLIENT_ID' must be private and have accessor methods. - line: 31
 - Variable 'SIGNUP_URL' must be private and have accessor methods. - line: 36
 - Variable 'REDIRECT_URL' must be private and have accessor methods. - line: 41
- Reliability
 - Close Resource
 - ExportUserServlet
 - Ensure that resources like this ServletOutputStream object are closed after use - line: 58
 - Design For Extension
 - ApiNinjaResult
 - Class 'ApiNinjaResult' looks like designed for extension (can be subclassed), but the method 'toString' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ApiNinjaResult' final or making the method 'toString' static/final/abstract/empty, or adding allowed annotation for the method. - line: 135
 - Auth
 - Class 'Auth' looks like designed for extension (can be subclassed), but the method 'init' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'Auth' final or making the method 'init' static/final/abstract/empty, or adding allowed annotation for the method. - line: 101
 - CognitoTokenHeader
 - Class 'CognitoTokenHeader' looks like designed for extension (can be subclassed), but the method 'getKid' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'CognitoTokenHeader' final or making the method 'getKid' static/final/abstract/empty, or adding allowed annotation for the method. - line: 12
 - Class 'CognitoTokenHeader' looks like designed for extension (can be subclassed), but the method 'getAlg' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'CognitoTokenHeader' final or making the method 'getAlg' static/final/abstract/empty, or adding allowed annotation for the method. - line: 16
 - ExerciseDbJson
 - Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'setGifUrl' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'setGifUrl' static/final/abstract/empty, or adding allowed annotation for the method. - line: 31
 - Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'getGifUrl' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'getGifUrl' static/final/abstract/empty, or adding allowed annotation for the method. - line: 35
 - Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the

method 'setInstructions' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'setInstructions' static/final/abstract/empty, or adding allowed annotation for the method. - line: 39

- Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'getInstructions' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'getInstructions' static/final/abstract/empty, or adding allowed annotation for the method. - line: 43
- Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'setSecondaryMuscles' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'setSecondaryMuscles' static/final/abstract/empty, or adding allowed annotation for the method. - line: 47
- Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'getSecondaryMuscles' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'getSecondaryMuscles' static/final/abstract/empty, or adding allowed annotation for the method. - line: 51
- Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'setName' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'setName' static/final/abstract/empty, or adding allowed annotation for the method. - line: 55
- Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'getName' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'getName' static/final/abstract/empty, or adding allowed annotation for the method. - line: 59
- Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'setEquipment' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'setEquipment' static/final/abstract/empty, or adding allowed annotation for the method. - line: 63
- Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'getEquipment' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'getEquipment' static/final/abstract/empty, or adding allowed annotation for the method. - line: 67
- Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'setId' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'setId' static/final/abstract/empty, or adding allowed annotation for the method. - line: 71
- Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'getId' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'getId' static/final/abstract/empty, or adding allowed annotation

for the method. - line: 75

- Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'setBodyPart' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'setBodyPart' static/final/abstract/empty, or adding allowed annotation for the method. - line: 79
- Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'getBodyPart' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'getBodyPart' static/final/abstract/empty, or adding allowed annotation for the method. - line: 83
- Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'setTarget' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'setTarget' static/final/abstract/empty, or adding allowed annotation for the method. - line: 87
- Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'getTarget' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'getTarget' static/final/abstract/empty, or adding allowed annotation for the method. - line: 91
- Class 'ExerciseDbJson' looks like designed for extension (can be subclassed), but the method 'toString' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ExerciseDbJson' final or making the method 'toString' static/final/abstract/empty, or adding allowed annotation for the method. - line: 95
- HeightRecord
 - Class 'HeightRecord' looks like designed for extension (can be subclassed), but the method 'toString' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'HeightRecord' final or making the method 'toString' static/final/abstract/empty, or adding allowed annotation for the method. - line: 162
- HipRecord
 - Class 'HipRecord' looks like designed for extension (can be subclassed), but the method 'toString' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'HipRecord' final or making the method 'toString' static/final/abstract/empty, or adding allowed annotation for the method. - line: 149
- InitialPropertiesServlet
 - Class 'InitialPropertiesServlet' looks like designed for extension (can be subclassed), but the method 'init' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'InitialPropertiesServlet' final or making the method 'init' static/final/abstract/empty, or adding allowed annotation for the method. - line: 29
- Keys
 - Class 'Keys' looks like designed for extension (can be subclassed), but the method 'getKeys' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'Keys' final or making the method

'getKeys' static/final/abstract/empty, or adding allowed annotation for the method. - line: 11

- KeysItem
 - Class 'KeysItem' looks like designed for extension (can be subclassed), but the method 'getKty' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'KeysItem' final or making the method 'getKty' static/final/abstract/empty, or adding allowed annotation for the method. - line: 24
 - Class 'KeysItem' looks like designed for extension (can be subclassed), but the method 'getE' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'KeysItem' final or making the method 'getE' static/final/abstract/empty, or adding allowed annotation for the method. - line: 28
 - Class 'KeysItem' looks like designed for extension (can be subclassed), but the method 'getUse' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'KeysItem' final or making the method 'getUse' static/final/abstract/empty, or adding allowed annotation for the method. - line: 32
 - Class 'KeysItem' looks like designed for extension (can be subclassed), but the method 'getKid' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'KeysItem' final or making the method 'getKid' static/final/abstract/empty, or adding allowed annotation for the method. - line: 36
 - Class 'KeysItem' looks like designed for extension (can be subclassed), but the method 'getAlg' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'KeysItem' final or making the method 'getAlg' static/final/abstract/empty, or adding allowed annotation for the method. - line: 40
 - Class 'KeysItem' looks like designed for extension (can be subclassed), but the method 'getN' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'KeysItem' final or making the method 'getN' static/final/abstract/empty, or adding allowed annotation for the method. - line: 44
- LogIn
 - Class 'LogIn' looks like designed for extension (can be subclassed), but the method 'init' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'LogIn' final or making the method 'init' static/final/abstract/empty, or adding allowed annotation for the method. - line: 43
- RequestUser
 - Class 'RequestUser' looks like designed for extension (can be subclassed), but the method 'getClasses' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'RequestUser' final or making the method 'getClasses' static/final/abstract/empty, or adding allowed annotation for the method. - line: 12
- SignUpServlet
 - Class 'SignUpServlet' looks like designed for extension (can be subclassed), but the method 'init' does not have javadoc that explains how to do that safely. If class is not

designed for extension consider making the class 'SignUpServlet' final or making the method 'init' static/final/abstract/empty, or adding allowed annotation for the method. - line: 43

- TokenResponse
 - Class 'TokenResponse' looks like designed for extension (can be subclassed), but the method 'getAccessToken' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'TokenResponse' final or making the method 'getAccessToken' static/final/abstract/empty, or adding allowed annotation for the method. - line: 21
 - Class 'TokenResponse' looks like designed for extension (can be subclassed), but the method 'getRefreshToken' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'TokenResponse' final or making the method 'getRefreshToken' static/final/abstract/empty, or adding allowed annotation for the method. - line: 25
 - Class 'TokenResponse' looks like designed for extension (can be subclassed), but the method 'getIdToken' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'TokenResponse' final or making the method 'getIdToken' static/final/abstract/empty, or adding allowed annotation for the method. - line: 29
 - Class 'TokenResponse' looks like designed for extension (can be subclassed), but the method 'getTokenType' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'TokenResponse' final or making the method 'getTokenType' static/final/abstract/empty, or adding allowed annotation for the method. - line: 33
 - Class 'TokenResponse' looks like designed for extension (can be subclassed), but the method 'getExpiresIn' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'TokenResponse' final or making the method 'getExpiresIn' static/final/abstract/empty, or adding allowed annotation for the method. - line: 37
- User
 - Class 'User' looks like designed for extension (can be subclassed), but the method 'getAgeNumber' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'User' final or making the method 'getAgeNumber' static/final/abstract/empty, or adding allowed annotation for the method. - line: 330
 - Class 'User' looks like designed for extension (can be subclassed), but the method 'toString' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'User' final or making the method 'toString' static/final/abstract/empty, or adding allowed annotation for the method. - line: 338
- UsersHTML
 - Class 'UsersHTML' looks like designed for extension (can be subclassed), but the method 'getMessage' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'UsersHTML' final or making the method 'getMessage' static/final/abstract/empty, or adding allowed annotation for the method. - line: 17
- UsersJSON
 - Class 'UsersJSON' looks like designed for extension (can be subclassed), but the

method 'getMessage' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'UsersJSON' final or making the method 'getMessage' static/final/abstract/empty, or adding allowed annotation for the method. - line: 20

- UsersText
 - Class 'UsersText' looks like designed for extension (can be subclassed), but the method 'getMessage' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'UsersText' final or making the method 'getMessage' static/final/abstract/empty, or adding allowed annotation for the method. - line: 18
- WaistRecord
 - Class 'WaistRecord' looks like designed for extension (can be subclassed), but the method 'toString' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'WaistRecord' final or making the method 'toString' static/final/abstract/empty, or adding allowed annotation for the method. - line: 150
- WeightRecord
 - Class 'WeightRecord' looks like designed for extension (can be subclassed), but the method 'toString' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'WeightRecord' final or making the method 'toString' static/final/abstract/empty, or adding allowed annotation for the method. - line: 149
- Magic Number
 - ExportUserServlet
 - '4096' is a magic number. - line: 63
 - '4096' is a magic number. - line: 64
 - '4096' is a magic number. - line: 65
 - HealthCalculations
 - '100' is a magic number. - line: 66
 - '0.23' is a magic number. - line: 103
 - '1.20' is a magic number. - line: 103
 - '-16.2' is a magic number. - line: 104
 - '-5.4' is a magic number. - line: 105
 - '88.362' is a magic number. - line: 126
 - '13.397' is a magic number. - line: 126
 - '5.677' is a magic number. - line: 126
 - '4.799' is a magic number. - line: 126
 - '447.593' is a magic number. - line: 127
 - '4.330' is a magic number. - line: 127
 - '3.098' is a magic number. - line: 127
 - '9.247' is a magic number. - line: 127
 - '.5' is a magic number. - line: 147
 - '1.5' is a magic number. - line: 153
 - '0.393701' is a magic number. - line: 171
 - '60' is a magic number. - line: 176
 - '50' is a magic number. - line: 176

- '2.3' is a magic number. - line: 176
 - '60' is a magic number. - line: 177
 - '2.3' is a magic number. - line: 177
 - '45.5' is a magic number. - line: 177
 - '0.267' is a magic number. - line: 197
 - '19.2' is a magic number. - line: 197
 - '0.407' is a magic number. - line: 197
 - '0.473' is a magic number. - line: 198
 - '0.252' is a magic number. - line: 198
 - '48.3' is a magic number. - line: 198
 - UsersHTML
 - '200' is a magic number. - line: 82
 - UsersJSON
 - '200' is a magic number. - line: 31
 - '500' is a magic number. - line: 36
 - UsersText
 - '200' is a magic number. - line: 30
- Usability
 - Constant Name
 - PropertiesLoader
 - Name 'logger' must match pattern `^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$`. - line: 21
 - Hidden Field
 - ApiNinjaResult
 - 'difficulty' hides a field. - line: 32
 - 'instructions' hides a field. - line: 50
 - 'muscle' hides a field. - line: 68
 - 'name' hides a field. - line: 86
 - 'equipment' hides a field. - line: 104
 - 'type' hides a field. - line: 122
 - ExerciseDbJson
 - 'gifUrl' hides a field. - line: 31
 - 'instructions' hides a field. - line: 39
 - 'secondaryMuscles' hides a field. - line: 47
 - 'name' hides a field. - line: 55
 - 'equipment' hides a field. - line: 63
 - 'id' hides a field. - line: 71
 - 'bodyPart' hides a field. - line: 79
 - 'target' hides a field. - line: 87
 - GenericDao
 - 'paramClass' hides a field. - line: 32
 - HealthCalculations
 - 'user' hides a field. - line: 30
 - HeightRecord
 - 'height' hides a field. - line: 41

- 'height' hides a field. - line: 51
- 'user' hides a field. - line: 51
- 'height' hides a field. - line: 63
- 'id' hides a field. - line: 63
- 'user' hides a field. - line: 63
- 'user' hides a field. - line: 69
- 'height' hides a field. - line: 69
- 'entryDate' hides a field. - line: 69
- 'id' hides a field. - line: 83
- 'entryDate' hides a field. - line: 83
- 'user' hides a field. - line: 83
- 'height' hides a field. - line: 83
- 'id' hides a field. - line: 104
- 'user' hides a field. - line: 122
- 'height' hides a field. - line: 140
- 'entryDate' hides a field. - line: 158
- HipRecord
 - 'hip' hides a field. - line: 41
 - 'user' hides a field. - line: 45
 - 'hip' hides a field. - line: 45
 - 'user' hides a field. - line: 50
 - 'hip' hides a field. - line: 50
 - 'id' hides a field. - line: 50
 - 'entryDate' hides a field. - line: 56
 - 'user' hides a field. - line: 56
 - 'hip' hides a field. - line: 56
 - 'entryDate' hides a field. - line: 70
 - 'user' hides a field. - line: 70
 - 'id' hides a field. - line: 70
 - 'hip' hides a field. - line: 70
 - 'id' hides a field. - line: 91
 - 'user' hides a field. - line: 109
 - 'hip' hides a field. - line: 127
 - 'entryDate' hides a field. - line: 145
- User
 - 'id' hides a field. - line: 73
 - 'userEmail' hides a field. - line: 82
 - 'userEmail' hides a field. - line: 92
 - 'id' hides a field. - line: 92
 - 'gender' hides a field. - line: 106
 - 'age' hides a field. - line: 106
 - 'firstName' hides a field. - line: 106
 - 'lastName' hides a field. - line: 106
 - 'userEmail' hides a field. - line: 106

- 'id' hides a field. - line: 128
- 'userEmail' hides a field. - line: 146
- 'firstName' hides a field. - line: 164
- 'lastName' hides a field. - line: 182
- 'gender' hides a field. - line: 200
- 'age' hides a field. - line: 218
- 'activityLevel' hides a field. - line: 236
- 'weightRecords' hides a field. - line: 272
- 'heightRecord' hides a field. - line: 290
- 'hipRecord' hides a field. - line: 308
- 'waistRecord' hides a field. - line: 326
- WaistRecord
 - 'waist' hides a field. - line: 42
 - 'user' hides a field. - line: 46
 - 'waist' hides a field. - line: 46
 - 'id' hides a field. - line: 51
 - 'waist' hides a field. - line: 51
 - 'user' hides a field. - line: 51
 - 'user' hides a field. - line: 57
 - 'entryDate' hides a field. - line: 57
 - 'waist' hides a field. - line: 57
 - 'entryDate' hides a field. - line: 71
 - 'waist' hides a field. - line: 71
 - 'id' hides a field. - line: 71
 - 'user' hides a field. - line: 71
 - 'id' hides a field. - line: 92
 - 'user' hides a field. - line: 110
 - 'waist' hides a field. - line: 128
 - 'entryDate' hides a field. - line: 146
- WeightRecord
 - 'weight' hides a field. - line: 41
 - 'weight' hides a field. - line: 45
 - 'user' hides a field. - line: 45
 - 'user' hides a field. - line: 50
 - 'weight' hides a field. - line: 50
 - 'id' hides a field. - line: 50
 - 'weight' hides a field. - line: 56
 - 'entryDate' hides a field. - line: 56
 - 'user' hides a field. - line: 56
 - 'user' hides a field. - line: 70
 - 'entryDate' hides a field. - line: 70
 - 'weight' hides a field. - line: 70
 - 'id' hides a field. - line: 70
 - 'id' hides a field. - line: 91

- 'user' hides a field. - line: 109
 - 'weight' hides a field. - line: 127
 - 'entryDate' hides a field. - line: 145
 - Local Variable Name
 - AccountInsertUpdate
 - Name 'activity_level' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 43
 - Member Name
 - Auth
 - Name 'CLIENT_ID' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 62
 - Name 'CLIENT_SECRET' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 67
 - Name 'OAUTH_URL' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 72
 - Name 'LOGIN_URL' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 77
 - Name 'REDIRECT_URL' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 82
 - Name 'REGION' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 87
 - Name 'POOL_ID' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 92
 - KeyItem
 - Name 'E' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 10
 - Name 'N' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 22
 - Naming - Suspicious constant field name
 - Auth
 - The field name indicates a constant but its modifiers do not - line: 62
 - The field name indicates a constant but its modifiers do not - line: 67
 - The field name indicates a constant but its modifiers do not - line: 72
 - The field name indicates a constant but its modifiers do not - line: 77
 - The field name indicates a constant but its modifiers do not - line: 82
 - The field name indicates a constant but its modifiers do not - line: 87
 - The field name indicates a constant but its modifiers do not - line: 92
 - KeyItem
 - The field name indicates a constant but its modifiers do not - line: 10
 - The field name indicates a constant but its modifiers do not - line: 22
 - LogIn
 - The field name indicates a constant but its modifiers do not - line: 31
 - The field name indicates a constant but its modifiers do not - line: 36
 - The field name indicates a constant but its modifiers do not - line: 41
 - SignUpServlet
 - The field name indicates a constant but its modifiers do not - line: 31
 - The field name indicates a constant but its modifiers do not - line: 36
 - The field name indicates a constant but its modifiers do not - line: 41
 - Parameter Assignment
 - HealthCalculations
 - Assignment of parameter 'formulaNumber' is not allowed. - line: 230
 - Assignment of parameter 'formulaNumber' is not allowed. - line: 232
 - Assignment of parameter 'formulaNumber' is not allowed. - line: 234
 - Assignment of parameter 'formulaNumber' is not allowed. - line: 238

- Assignment of parameter 'formulaNumber' is not allowed. - line: 241
- Parameter Name
 - User
 - Name 'entry_date' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 254
- Static Variable Name
 - LogIn
 - Name 'CLIENT_ID' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 31
 - Name 'LOGIN_URL' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 36
 - Name 'REDIRECT_URL' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 41
 - SignUpServlet
 - Name 'CLIENT_ID' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 31
 - Name 'SIGNUP_URL' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 36
 - Name 'REDIRECT_URL' must match pattern '^[a-z][a-zA-Z0-9]*\$'. - line: 41