SUSTech Southern University of Science and Technology

ModelMapper

Group Name:

404notfound

Team members:

刘通 11910903

赵晓蕾 11910937

段轶 11911133

陈梓涵 11910507

张庭境 11911919

Introduction

modelmapper

Simple, Intelligent, Object Mapping.

SUSTech Southern University of Science and Technology

- **Why Map?**

  Applications often consist of similar but different object models, where the data in two models may be similar but the structure and concerns of the models are different. Object mapping makes it easy to convert one model to another, allowing separate models to remain segregated.

- **Why ModelMapper?**

  ModelMapper is to make object mapping easy, by automatically determining how one object model maps to another, based on conventions, in the same way that a human would - while providing a simple, refactoring-safe API for handling specific use cases.

Timeline

SUSTech Southern University of Science and Technology

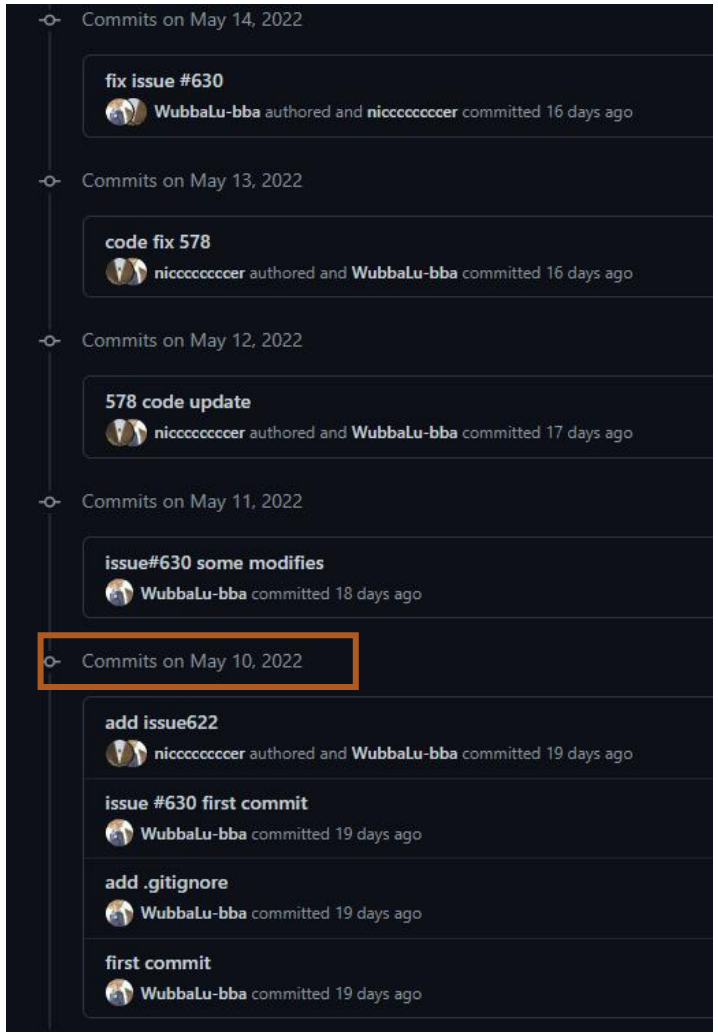
Week	Issues Implemented
Week 5-7	Build the project in our computer and do some tests. Recognize the design and structure of the project.
Week 8	Fix the issue#572 Ability to create an empty named typemap Reproduce the issue#660 Failed to instantiate instance of destination java.util.UUID. Ensure that java.util.UUID has a non-private no-argument constructor.
Week 9	Fix the issue#660 Failed to instantiate instance of destination java.util.UUID. Ensure that java.util.UUID has a non-private no-argument constructor. Fix the issue#633 ModelMapper not skip a object field.
Week 10	Fix the issue#639 MappingContext.getDestination() always null Reproduce the issue#278 Map<String, List<SomeObject>>: not mapped

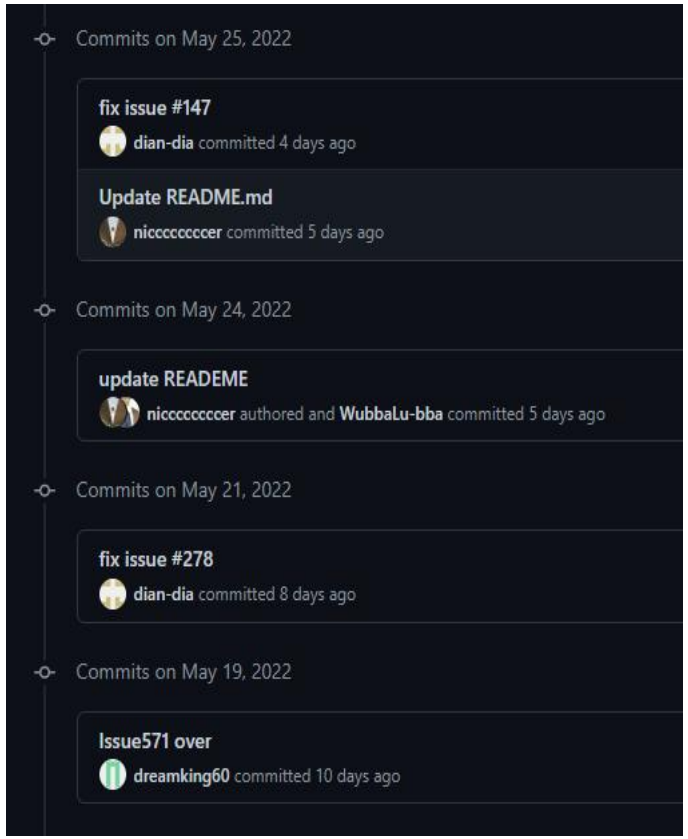
Timeline

SUSTech Southern University of Science and Technology

Week	Issues Implemented
Week 12	Try to reproduce the issue#622 which is addConverter not working. Reproduce the issue#630 which is mapping with same class not working.
Week 13	Fix the issue#630 so that mapping with same class can work. Reproduce and fix the issue#578 which is conflicts of explicit skipping of nested value. Reproduce the issue#571 which is add dynamic properties to MappingContext.
Week 14	Reproduce and fix the issue#278 Map<String, List<SomeObject>>: not mapped. Fix the issue#571 which enhances to add dynamic properties to MappingContext.
Week 15	Fix the issue#164 Cyclic mapping issue in projection. Fix the issue#147 Unexpected mapping with some name.

History


Commits on May 14, 2022
fix issue #630
WubbaLu-bba authored and niccccccccer committed 16 days ago
Commits on May 13, 2022
code fix 578
niccccccccer authored and WubbaLu-bba committed 16 days ago
Commits on May 12, 2022
578 code update
niccccccccer authored and WubbaLu-bba committed 17 days ago
Commits on May 11, 2022
issue#630 some modifies
WubbaLu-bba committed 18 days ago
Commits on May 10, 2022
add issue622
niccccccccer authored and WubbaLu-bba committed 19 days ago
issue #630 first commit
WubbaLu-bba committed 19 days ago
add .gitignore
WubbaLu-bba committed 19 days ago
first commit
WubbaLu-bba committed 19 days ago


Commits on May 25, 2022
fix issue #147
dian-dia committed 4 days ago
Update README.md
niccccccccer committed 5 days ago
Commits on May 24, 2022
update README
niccccccccer authored and WubbaLu-bba committed 5 days ago
Commits on May 21, 2022
fix issue #278
dian-dia committed 8 days ago
Commits on May 19, 2022
Issue571 over
dreamking60 committed 10 days ago


Commits on May 29, 2022
add midterm progress
WubbaLu-bba committed 13 minutes ago
Update ConditionTest.java
YiYim committed 3 hours ago
Update and rename Obj.java to ConditionTest.java
YiYim committed 3 hours ago
Create ObjTest.java
YiYim committed 3 hours ago
Update and rename 1 to Obj.java
YiYim committed 3 hours ago
Create 1
YiYim committed 3 hours ago
Issue 571
dreamking60 committed 7 hours ago
Update README.md
WubbaLu-bba committed 17 hours ago
Commits on May 27, 2022
Update README.md
niccccccccer committed 3 days ago
Commits on May 26, 2022
update README
niccccccccer committed 4 days ago

Important issues implemented

SUSTech Southern University of Science and Technology

- #630: When map a class with property in the same class, the mapping will not run correctly, so when check the mapping, should not record the type of the class, we add a condition to avoid this happen：

```
//CS304 Issue link: https://github.com/modelmapper/modelmapper/issues/630
/**
 * Matches the {@code destinationTypeInfo}'s mutator hierarchy hierarchy to the
 * {@code sourceTypeInfo}'s accessor hierarchy.
 */
private void matchDestination(TypeInfo<?> destinationTypeInfo) {
  if (!(propertyNameInfo.getDestinationProperties().size() == 0)){
    destinationTypes.add(destinationTypeInfo.getType());
  }
```

Important issues implemented

SUSTech Southern University of Science and Technology

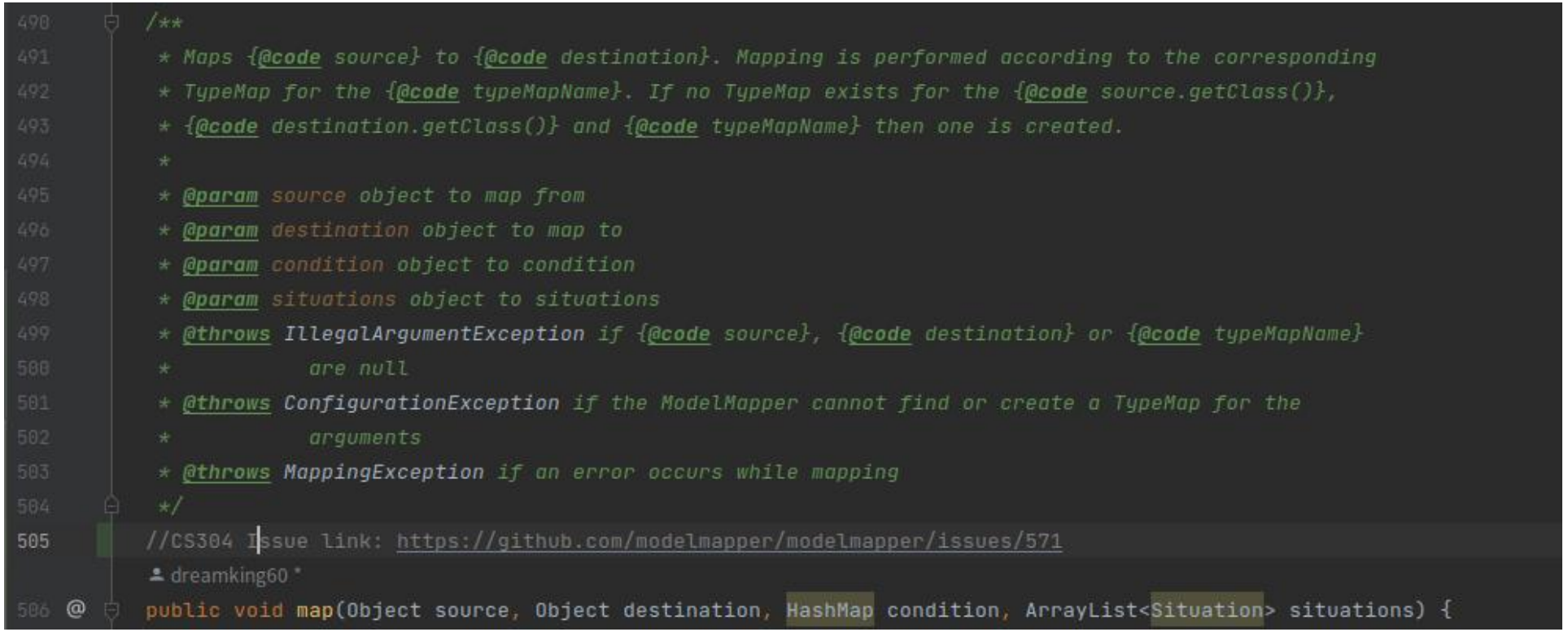
- #578: When skip a map of an object property, the map cannot recursively cover the property inside it, so as design, it should throw exception, but in this case it did not, so we fix the exception：

```java
//CS304 Issue link: https://github.com/modelmapper/modelmapper/issues/578
/**
 * Skip the mapping from the source property of sourceGetter
 * to the destination property of destinationSetter
 * V - the value type of the setter
 * @param sourceGetter - a method reference to source getter
 * @param destinationSetter - a method reference to destination setter
 */
@Override
public <V> void skip(SourceGetter<S> sourceGetter, DestinationSetter<D, V> destinationSetter) {
  options.skipType = 1;
  visitSource(sourceGetter);
  visitDestination(destinationSetter);
  skipMapping(collector.collect());
  collector.reset();
}
```

Important issues implemented

SUSTech Southern University of Science and Technology

- #571: For a condition mapping, it can only add conditions one by one, cannot have an elegant way to decide which properties is needed to map, so in this issue, we add this option：



```
490     /**
491      * Maps {@code source} to {@code destination}. Mapping is performed according to the corresponding
492      * TypeMap for the {@code typeMapName}. If no TypeMap exists for the {@code source.getClass()},
493      * {@code destination.getClass()} and {@code typeMapName} then one is created.
494      *
495      * @param source object to map from
496      * @param destination object to map to
497      * @param condition object to condition
498      * @param situations object to situations
499      * @throws IllegalArgumentException if {@code source}, {@code destination} or {@code typeMapName}
500      *           are null
501      * @throws ConfigurationException if the ModelMapper cannot find or create a TypeMap for the
502      *           arguments
503      * @throws MappingException if an error occurs while mapping
504      */
505     //CS304 Issue link: https://github.com/modelmapper/modelmapper/issues/571
        dreamking60 *
506 @   public void map(Object source, Object destination, HashMap condition, ArrayList<Situation> situations) {
```

Important issues implemented

SUSTech Southern University of Science and Technology

- #571:

```
505   //CS304 Issue link: https://github.com/modelmapper/modelmapper/issues/571
      dreamking60 *
506 @ public void map(Object source, Object destination, HashMap condition, ArrayList<Situation> situations) {
507     Assert.notNull(source, parameterName: "source");
508     Assert.notNull(destination, parameterName: "destination");
509     Assert.isTrue(!condition.isEmpty());
510
511     Iterator<Situation> iterator = situations.iterator();
512     Situation situation;
513
514     int check = 0;
515
516     while (iterator.hasNext()) {
517       situation = iterator.next();
518       if (situation.getSituation().equals(condition)) {
519         check = 1;
520         mapInternal(source, destination, destinationType: null, situation.getWay().getName());
521         break;
522       }
523     }
524     if (check == 0) {
525       System.out.println("No typeMap match the situation!");
526     }
527
528 }
```

Important issues implemented

SUSTech Southern University of Science and Technology

- #164: For cyclic mapping, which is about a map with a class have property in this class, like nested attribute, so in design, I make deep map in the first time, then in the circle do the shallow map:

```
for (Map.Entry<String, Mutator> entry : destinationTypeInfo.getMutators().entrySet()) {
  //CS304 Issue link: https://github.com/modelmapper/modelmapper/issues/164
  propertyNameInfo.pushDestination(entry.getKey(), entry.getValue());
  String destPath = Strings.join(propertyNameInfo.getDestinationProperties());
  Mutator mutator = entry.getValue();


  destinationTypes.remove(destinationTypeInfo.getType());
  errors.throwConfigurationExceptionIfErrorsExist();
```

Important issues implemented

SUSTech Southern University of Science and Technology

- #278: For mapping Map<String, List<SomeObject>>, it cannot map properly. So in the issue, we modify a condition when mapping:

```
//CS304 Issue link: https://github.com/modelmapper/modelmapper/issues/278
/**
 * Performs mapping using a TypeMap if one exists, else a converter if one applies, else a newly
 * created TypeMap. Recursive entry point.
 */
@Override
@SuppressWarnings({"unchecked", "checkstyle:NeedBraces", "checkstyle:WhitespaceAround"})
public <S, D> D map(MappingContext<S, D> context) {
  MappingContextImpl<S, D> contextImpl = (MappingContextImpl<S, D>) context;


  if (context.getDestinationType().isAssignableFrom(context.getSourceType()) && typeMap != null)
    destination = typeMap(contextImpl, typeMap);
  else
```

Important issues implemented

SUSTech Southern University of Science and Technology

- #147: When there are more elements in class to be matched than mapping class, the value of the unneeded variable may be modified. So in the issue, we use the strict matching strategy:

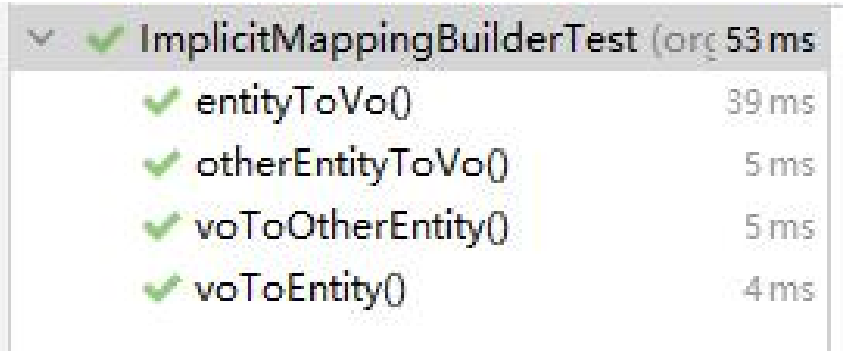
```
//CS304 Issue link: https://github.com/modelmapper/modelmapper/issues/147
/**
 * See {@link MatchingStrategies#STRICT}.
 *
 * @author Jonathan Halterman
 */
final class StrictMatchingStrategy implements MatchingStrategy {
  @Override
  public boolean isExact() { return true; }
```

```
//CS304 Issue link: https://github.com/modelmapper/modelmapper/issues/147
@Override
public boolean matches(PropertyNameInfo propertyNameInfo) {
  List<Tokens> sourceTokens = propertyNameInfo.getSourcePropertyTokens();
```

Testing techniques & test cases • #630

SUSTech Southern University of Science and Technology

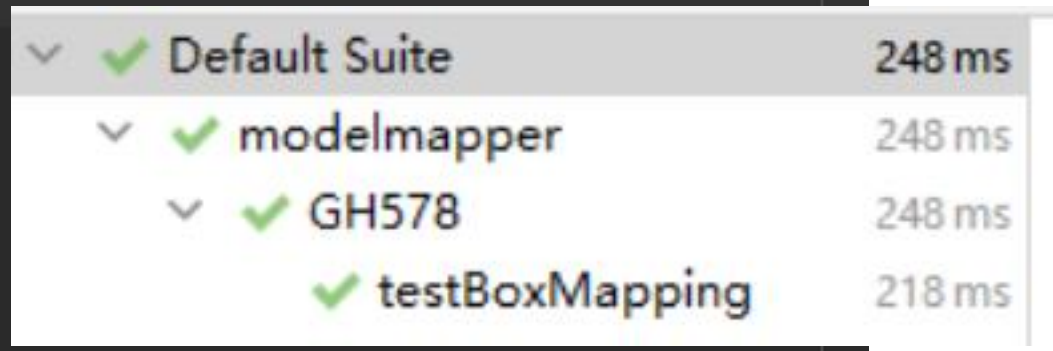
```java
//CS304 (manually written) Issue link: https://github.com/modelmapper/modelmapper/issues

public class ImplicitMappingBuilderTest {
    ModelMapper modelMapper = new ModelMapper();

    @Test
    public void entityToVo() throws Exception {
        Entity relation = Entity.builder().id(1).groupName("relation").build();
        Entity entity = Entity.builder().id(2).groupName("entity").relation(relation).bu
    @Test
    public void voToEntity() throws Exception {
        Vo vo = Vo.builder().id(2).relationId(1).build();

        Entity e = modelMapper.map(vo, Entity.class);
        Assertions.assertEquals(e.getId(), vo.getId());
        Assertions.assertEquals(e.getGroupName(), vo.getGroupName());
        Assertions.assertNotNull(e.getRelation());
        Assertions.assertEquals(e.getRelation().getId(), vo.getRelationId());
        Assertions.assertEquals(e.getRelation().getGroupName(), vo.getRelationGroupName(
    }
```



Testing techniques & test cases • #578

SUSTech Southern University of Science and Technology

```java
//CS304 (manually written) Issue link: https://github.com/modelmapper/modelmapper/issues/578
public void testBoxMapping() {
  TypeA typeA = new TypeA();
  TypeABox typeABox = new TypeABox();
  typeABox.setParam(CLASSIFIER.AAA);
  typeA.setBox(typeABox);
  TypeB typeB = new TypeB();
  TypeBBox typeBBox = new TypeBBox();
  typeBBox.setParam(CLASSIFIER.BBB);
  typeB.setBox(typeBBox);
  ModelMapper modelMapper = new ModelMapper();
  try {
    modelMapper.createTypeMap(TypeA.class, TypeB.class)
        .addMappings(mapper -> mapper.skip(TypeA::getBox, TypeB::setBox));
    fail();
  } catch (ConfigurationException e) {
    assertEquals(e.getErrorMessages().size(), expected: 1);
    assertEquals(e.getErrorMessages().iterator().next().getMessage(),
        expected: "Not able to skip box., because there are already nested properties are mapped: [box.param.]. "
            + "Do you skip the property after the implicit mappings mapped? "
            + "We recommended you to create an empty type map, and followed by addMappings and implicitMappings calls");
  }
  modelMapper.map(typeA, typeB);
}
}
```



Testing techniques & test cases • #571

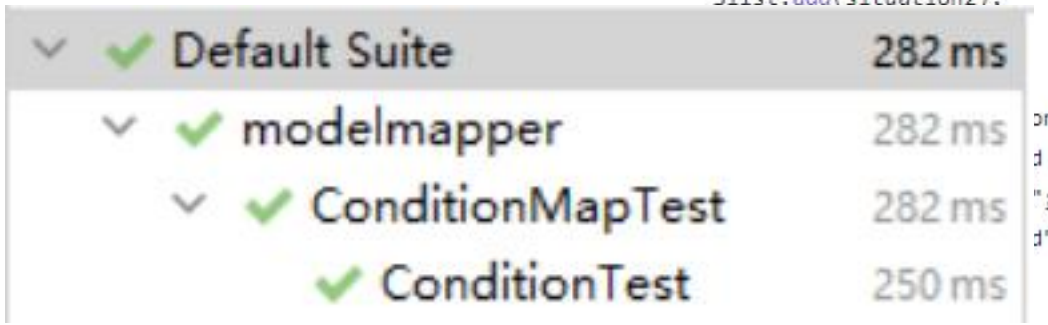
SUSTech Southern University of Science and Technology

```
//CS304 (manually written) Issue link: https://github.com/modelmapper/modelmapper/issues/571
```

```
TypeMap<Person, PersonDTO> map0 = modelMapper.createTypeMap(Person.class, PersonDTO.class, "map0").addMappings(mapper -> {
    mapper.map(Person::getLastName, PersonDTO::setSurName);
});
TypeMap<Person, PersonDTO> map1 = modelMapper.createTypeMap(Person.class, PersonDTO.class, "map1").addMappings(mapper -> {
    mapper.map(Person::getEmployer, PersonDTO::setEmployerName);
});
TypeMap<Person, PersonDTO> map2 = modelMapper.createTypeMap(Person.class, PersonDTO.class, "map2").addMappings(mapper -> {
    mapper.map(Person::getLastName, PersonDTO::setSurName);
    mapper.map(Person::getEmployer, PersonDTO::setEmployerName);
});
HashMap<String, Boolean> s0 = new HashMap<>();
s0.put("lastname",true);
HashMap<String, Boolean> s1 = new HashMap<>();
s1.put("employer", true);
HashMap<String, Boolean> s2 = new HashMap<>();
s2.put("lastname",true);
s2.put("employer", true);
```

```
ArrayList<Situation> Slist = new ArrayList<>();
Situation<Person, PersonDTO> situation1 = new Situation<>(s0, map0);
Situation<Person, PersonDTO> situation2 = new Situation<>(s1, map1);
Situation<Person, PersonDTO> situation3 = new Situation<>(s2, map2);
Slist.add(situation1);
Slist.add(situation2);
```

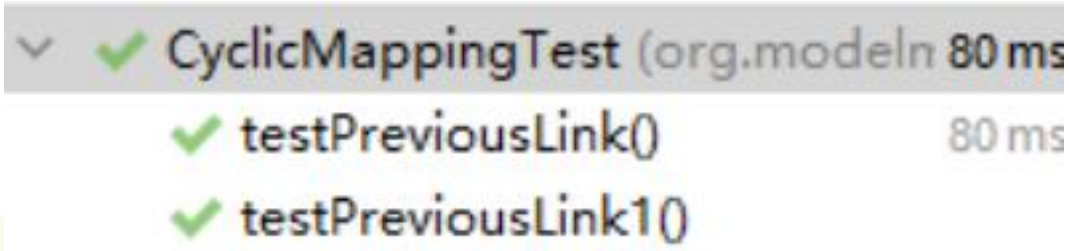
```
PersonDTO m0 = new PersonDTO();
modelMapper.map(person,m0,s0,Slist);
assertEquals(m0.surName,"Donald");
assertEquals(m0.employerName, null);
assertEquals(m0.firstName, "Mike");

PersonDTO m1 = new PersonDTO();
modelMapper.map(person,m1,s1,Slist);
assertEquals(m1.surName,null);
assertEquals(m1.employerName, "Donald Mike");
assertEquals(m1.firstName, "Mike");

PersonDTO m2 = new PersonDTO();
modelMapper.map(person,m2,s2,Slist);
assertEquals(m2.surName,"Donald");
assertEquals(m2.employerName, "Donald Mike");
assertEquals(m2.firstName, "Mike");
```



Testing techniques & test cases • #164

SUSTech Southern University of Science and Technology

```java
//CS304 (manually written) Issue link: https://github.com/modelmapper/modelmapper/issues/164
public void testPreviousLink() {

    ActionDTO dto = new ActionDTO();
    dto.setPreviousId(UUID.randomUUID());
    dto.setId(UUID.randomUUID());

    Action model = modelMapper.map(dto, Action.class);
    Assertions.assertInstanceOf(Action.class, model);
    Assertions.assertEquals(model.getId(), dto.getId());
    Assertions.assertNotNull(model.getPrevious()); // fails here when strict
    Assertions.assertInstanceOf(Action.class, model.getPrevious());
    Assertions.assertEquals(model.getPrevious().getId(), dto.getPreviousId()); //
}
@Test
public void testPreviousLink1() {

    ActionDTO dto = new ActionDTO();
    dto.setPreviousId(UUID.randomUUID());
    Action model = modelMapper.map(dto, Action.class);
    Assertions.assertInstanceOf(Action.class, model);
    Assertions.assertEquals(model.getId(), dto.getId());
    Assertions.assertNull(dto.getId());
    Assertions.assertEquals(model.getPrevious().getId(), dto.getPreviousId());
}
```



Testing techniques & test cases • #278

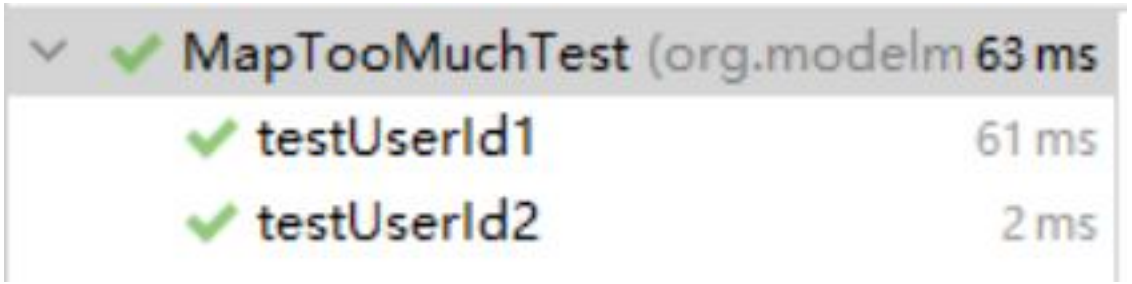
SUSTech Southern University of Science and Technology

```java
//CS304 (manually written) Issue link: https://github.com/modelmapper/modelmapper/issues/278
@Test
public void testMapToDto() {
    A source = new A();
    source.setAuthor("Author");
    Map<String, List<B>> map = new HashMap<>();
    List<B> list = new ArrayList<>();
    B b = new B();
    b.setTitle("a title");
    list.add(b);
    map.put("key 1", list);
    source.setMyMap(map);

    ADto aDto = modelMapper.map(source, ADto.class);

    Assert.assertEquals( expected: 1, aDto.getMyMap().size());
    Assert.assertEquals(aDto.getMyMap().get("key 1").get(0).getClass(), BDto.class);
}

@Before
public void init() {
    modelMapper = new ModelMapper();
    modelMapper.validate();
}
```

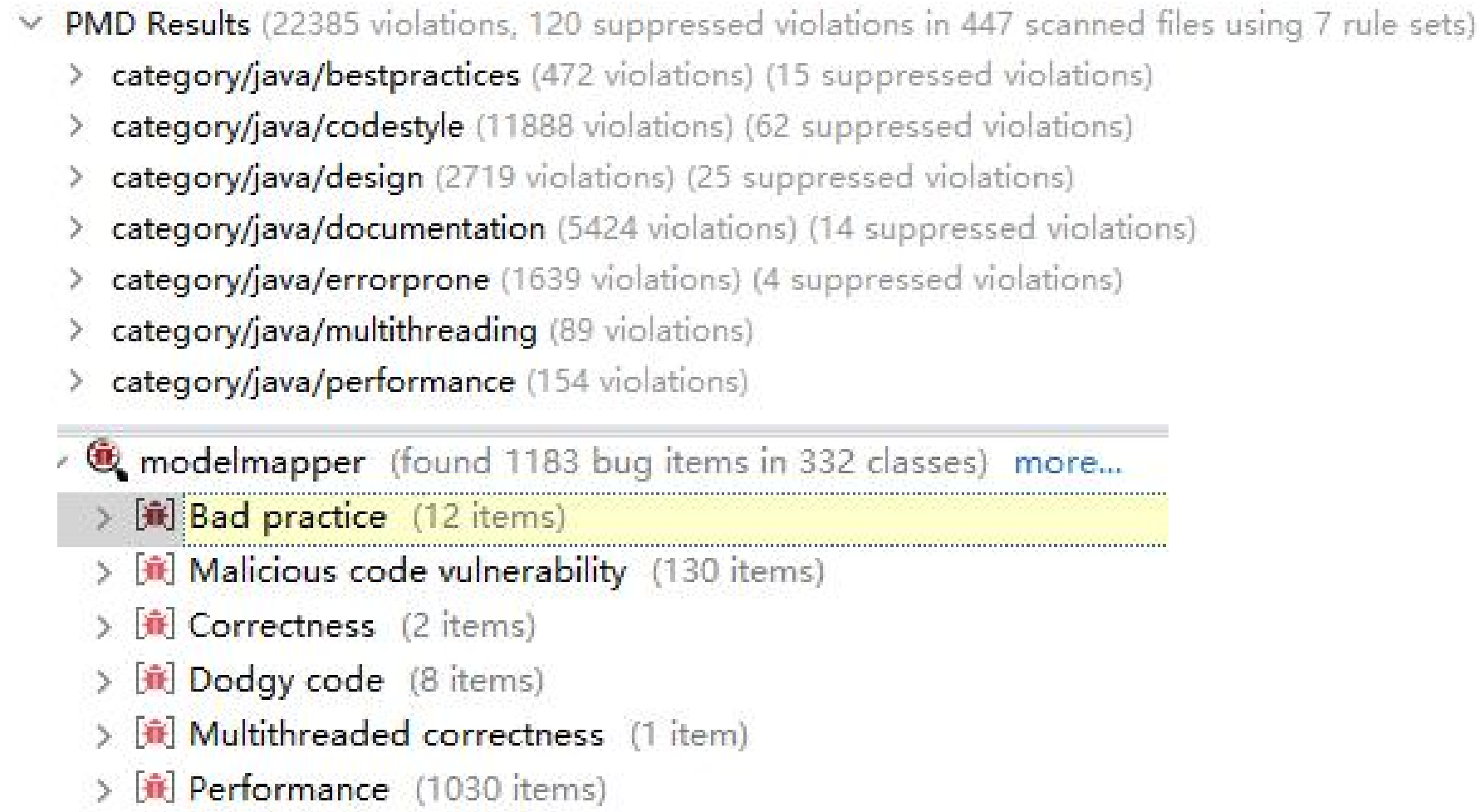
Testing techniques & test cases • #147

SUSTech Southern University of Science and Technology

```java
//CS304 (manually written) Issue link: https://github.com/modelmapper/modelmapper/issues/147
@Test
public void testUserId1() {
    NewUser newUser = new NewUser();
    newUser.setLogin("abcdefg");
    newUser.setBusinessId(1);

    ModelMapper mapper = new ModelMapper();
    mapper.getConfiguration().setMatchingStrategy(MatchingStrategies.STANDARD);
    User user = mapper.map(newUser, User.class);

    Assert.assertEquals(user.getId(), actual: 3);
    Assert.assertEquals(user.getLogin(), actual: "abcdefg");
    Assert.assertEquals(user.getBusinessId(), actual: 1);
}
//CS304 (manually written) Issue link: https://github.com/modelmapper/modelmapper/issues/147
@Test
public void testUserId2() {
    NewUser newUser = new NewUser();
    newUser.setLogin("abcdefg");
    newUser.setBusinessId(1);

    ModelMapper mapper = new ModelMapper();
    mapper.getConfiguration().setMatchingStrategy(MatchingStrategies.STRICT);
    User user = mapper.map(newUser, User.class);

    Assert.assertEquals(user.getId(), actual: 3);
    Assert.assertEquals(user.getLogin(), actual: "abcdefg");
    Assert.assertEquals(user.getBusinessId(), actual: 1);
}
```



Static analysis tool



Static analysis tool

SUSTech Southern University of Science and Technology

- Checkstyle found 69 item(s) in 1 file(s)
  - ModelMapper.java : 69 item(s)
    - 'package' should be separated from previous line. (16:1) [EmptyLineSeparator]
    - Extra separation in import group before 'java.lang.reflect.Type' (20:1) [CustomImportOrder]
    - Wrong lexicographical order for 'java.lang.reflect.Type' import. Should be before 'net.jodah.typetools.TypeResolver'. (20:1) [CustomImportOrder]
    - Wrong lexicographical order for 'java.util.ArrayList' import. Should be before 'net.jodah.typetools.TypeResolver'. (21:1) [CustomImportOrder]
    - Wrong lexicographical order for 'java.util.Collection' import. Should be before 'net.jodah.typetools.TypeResolver'. (22:1) [CustomImportOrder]
- Checkstyle found 2 item(s) in 1 file(s)
  - Provider.java : 2 item(s)
    - Javadoc tag '@param' should be preceded with an empty line. (24:0) [RequireEmptyLineBeforeBlockTagGroup]
    - Javadoc tag '@param' should be preceded with an empty line. (46:0) [RequireEmptyLineBeforeBlockTagGroup]
- Checkstyle found 12 item(s) in 1 file(s)
  - Situation.java : 12 item(s)
    - 'member def modifier' has incorrect indentation level 4, expected level should be 2. (15:5) [Indentation]
    - 'member def modifier' has incorrect indentation level 4, expected level should be 2. (16:5) [Indentation]
- Checkstyle found 32 item(s) in 1 file(s)
  - ImplicitMappingBuilder.java : 32 item(s)
    - 'package' should be separated from previous line. (16:1) [EmptyLineSeparator]
    - Extra separation in import group before 'org.modelmapper.Converter' (28:1) [CustomImportOrder]
    - Javadoc tag '@param' should be preceded with an empty line. (47:0) [RequireEmptyLineBeforeBlockTagGroup]
- Checkstyle found 52 item(s) in 1 file(s)
  - MappingEngineImpl.java : 52 item(s)
    - 'package' should be separated from previous line. (16:1) [EmptyLineSeparator]
    - Using the '.*' form of import should be avoided - org.modelmapper.*. (18:23) [AvoidStarImport]
    - Using the '.*' form of import should be avoided - org.modelmapper.spi.*. (24:27) [AvoidStarImport]

Results(scenarios)

SUSTech Southern University of Science and Technology

```java
public static class Vo {
    long id;
    String groupName;
    long relationId;
    String relationGroupName;
}
```

```java
public static class Entity {
    long id;
    String groupName;
    Entity relation;
}
```

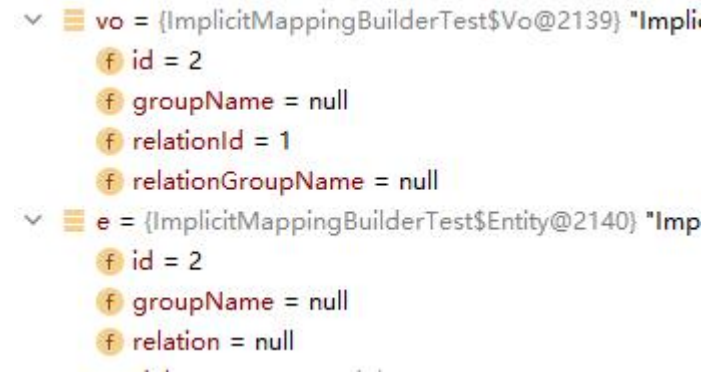
```java
public static class OtherEntity {
    long id;
    String groupName;
    DifferentEntity relation;

public static class DifferentEntity {
    long id;
    String groupName;
    OtherEntity relation;
}
```

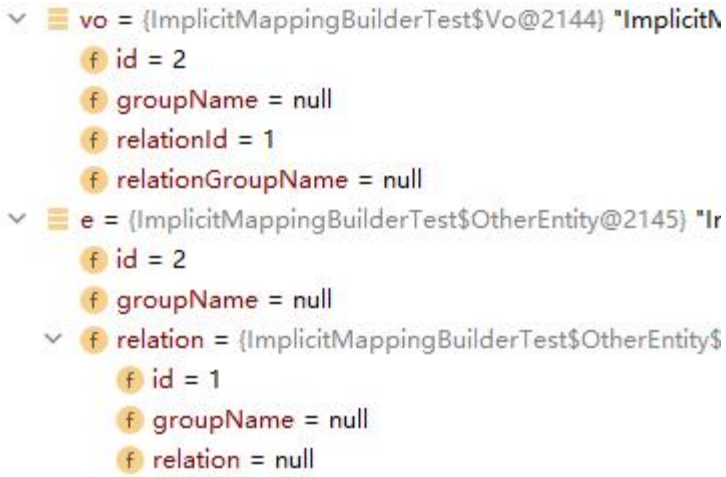
```java
@Test
public void voToOtherEntity() throws Exception {
    Vo vo = Vo.builder().id(2).relationId(1).build();
@Test
public void voToEntity() throws Exception {
    Vo vo = Vo.builder().id(2).relationId(1).build();
```

Results(scenarios)

SUSTech Southern University of Science and Technology

before

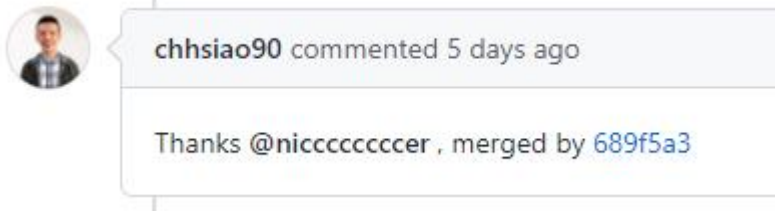


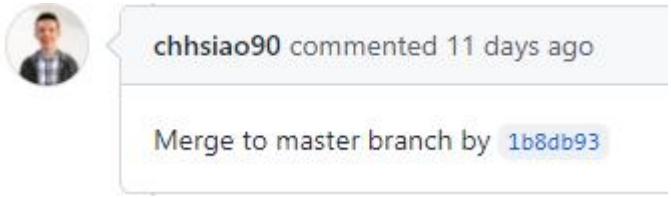


after

Conclusions

- After this project, our team have made great process in the model mapper, we have fixed 10 issues, both enhancements and bugs, and we contribute varieties of commits to the project and our work was appreciate by the developers.





Future Work

SUSTech Southern University of Science and Technology

- **What lessons did you learn from your project?**

  Help us use the version control tool git proficiently to collaborate to develop projects. And we learned the principles of automatically projecting and flattening complex models. Besides, we learned about fluent mapping API which is type-safe and refactoring-safe, using actual code, rather than string references, to map properties and values.

- **What was difficult?**

  How to locate the root of the problem according to the recurring scenario. The structure of the modelmapper is very complete, and it may be necessary to understand and modify the method according to the layers of nesting.

Future Work

SUSTech Southern University of Science and Technology

- **What do you wish you could have done (or done differently)?**

  For the nested values, the skipping method defined by the users may not work as we want. We want to solve this conflict, but the owner of the modelmapper do not want to solve it and think this is the limitation and can just throw exceptions.

- **How could your project be extended... what's next? Are there any interesting problems or questions that resulted from your work?**

  Similar but different object models often in many applications, where the data in two models may be similar but the structure and concerns of the models are different. So we can find more usage scenarios for this project and find the problems of this project in the test and solve them.

SUSTech Southern University of Science and Technology

THANKS