

Можно я тут API чуть-чуть поменяю?...

Дмитрий Цителов



О себе

Руководитель исследовательской
лаборатории dxLab, Devexperts

dxLab:

- Многопоточные алгоритмы
- Методы верификации многопоточных программ
- Инструменты анализа программ



“The secret to building large apps is never build large apps. Break your applications into small pieces. Then, assemble those testable, bite-sized pieces into your big application.”

— Justin Meyer

“Programming is like sex: one mistake and you have to support it for the rest of your life.”

— Michael Sinz

“I've finally learned what "upward compatible" means. It means we get to keep all our old mistakes.”

— Dennie Van Tassel

“The secret to building large apps is never build large apps. Break your applications into small pieces. Then, assemble those testable, bite-sized pieces into your big application.”

— Justin Meyer

“Programming is like sex: one mistake and you have to support it for the rest of your life.”

— Michael Sinz

“I've finally learned what "upward compatible" means. It means we get to keep all our old mistakes.”

— Dennie Van Tassel

“The secret to building large apps is never build large apps. Break your applications into small pieces. Then, assemble those testable, bite-sized pieces into your big application.”

— Justin Meyer

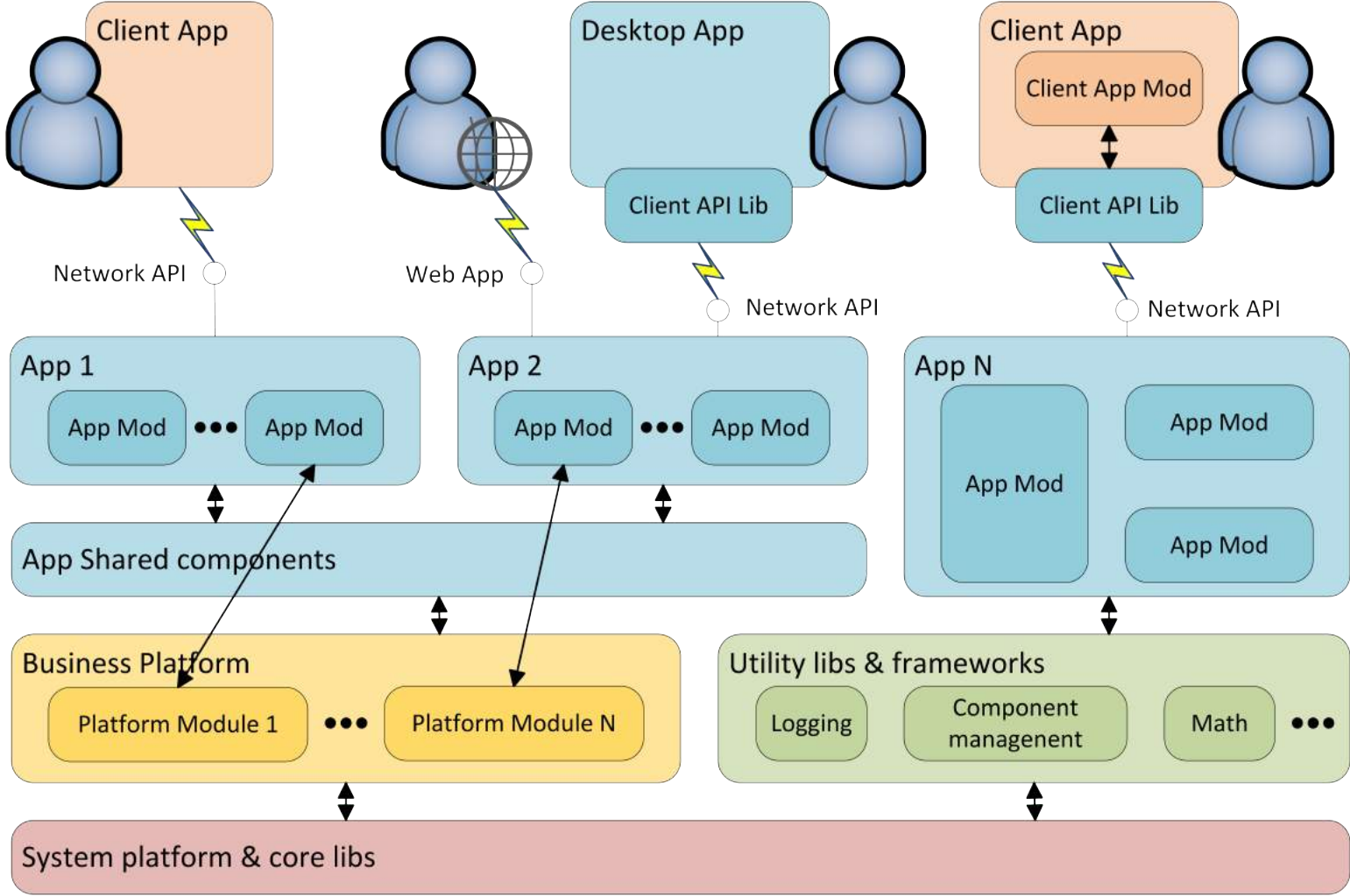
“Programming is like sex: one mistake and you have to support it for the rest of your life.”

— Michael Sinz

“I've finally learned what "upward compatible" means. It means we get to keep all our old mistakes.”

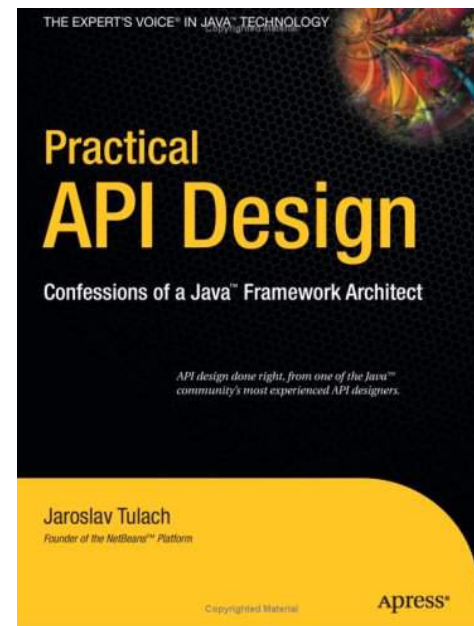
— Dennie Van Tassel

Модули и интерфейсы



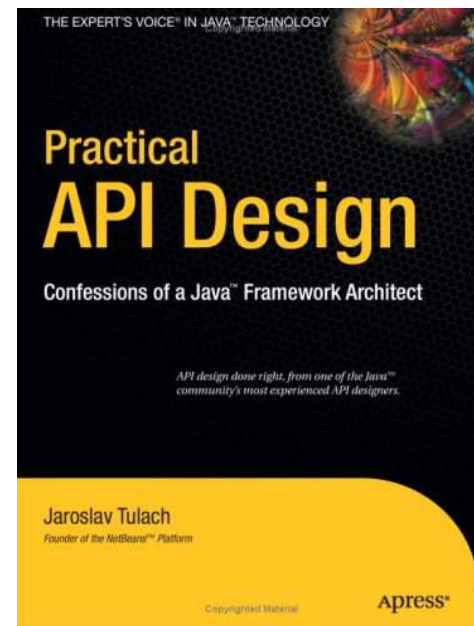
Разрабатываем библиотеки и пакеты

- Хорошее API - залог успеха
 - Логичная и удобная модель данных
 - Понятная и консистентная схема именования
 - Продуманные сценарии использования и расширения
 - Модульная архитектура
 - Стратегия развития и обеспечения обратной совместимости
- Документация
 - Чётко описанная модель и контракты
 - Документация изменений
- Сопровождение
 - Логичная политика версионирования, «дорожные карты»
 - Коммуникация с пользователями



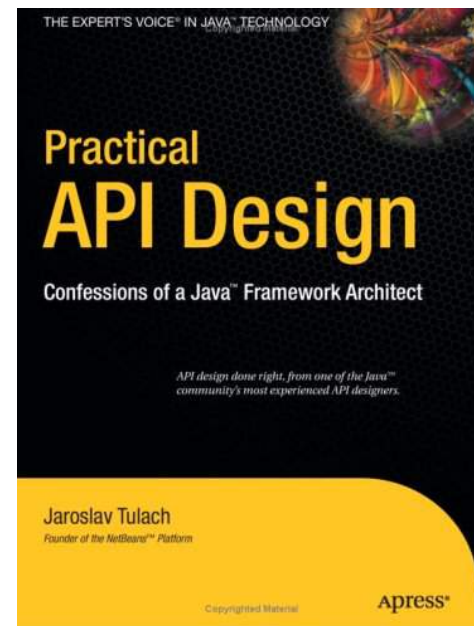
Разрабатываем библиотеки и пакеты

- Хорошее API - залог успеха
 - Логичная и удобная модель данных
 - Понятная и консистентная схема именования
 - Продуманные сценарии использования и расширения
 - Модульная архитектура
 - Стратегия развития и обеспечения обратной совместимости
- Документация
 - Чётко описанная модель и контракты
 - Документация изменений
- Сопровождение
 - Логичная политика версионирования, «дорожные карты»
 - Коммуникация с пользователями



Разрабатываем библиотеки и пакеты

- Хорошее API - залог успеха
 - Логичная и удобная модель данных
 - Понятная и консистентная схема именования
 - Продуманные сценарии использования и расширения
 - Модульная архитектура
 - Стратегия развития и обеспечения обратной совместимости
- Документация
 - Чётко описанная модель и контракты
 - Документация изменений
- Сопровождение
 - Логичная политика версионирования, «дорожные карты»
 - Коммуникация с пользователями



Проблемы поддержки общего API

- Больше пользователей - больше изменений (разнонаправленных)
- Нарастает энтропия, архитектура “плывёт”
 - Разрастаются списки параметров
 - Появляются сложные скрытые протоколы

Нужно что-то делать...



Проблемы поддержки общего API

- Больше пользователей - больше изменений (разнонаправленных)
- Нарастает энтропия, архитектура “плывёт”
 - Разрастаются списки параметров
 - Появляются сложные скрытые протоколы



Нужно что-то делать... О! Рефакторинг!

Проблемы поддержки общего API

- Больше пользователей - больше изменений (разнонаправленных)
- Нарастает энтропия, архитектура “плывёт”
 - Разрастаются списки параметров
 - Появляются сложные скрытые протоколы



Нужно что-то делать... О! Рефакторинг!

API хочется менять!

Проблемы поддержки общего API

- Больше пользователей - больше изменений (разнонаправленных)
- Нарастает энтропия, архитектура “плывёт”
 - Разрастаются списки параметров
 - Появляются сложные скрытые протоколы



Нужно что-то делать... О! Рефакторинг!

API хочется менять!

... А как же обратная совместимость?

Эволюция API и обратная совместимость

“Что написано пером, того не вырубишь топором”

- Параллельно строим новый мир
- Старое → `@deprecated` и ждём
- Объём поддерживаемого кода непрерывно растёт



Эволюция API и обратная совместимость

“Что написано пером, того не вырубишь топором”

- Параллельно строим новый мир
- Старое → `@deprecated` и ждём
- Объём поддерживаемого кода непрерывно растёт



“Кто не спрятался - я не виноват”

- Всё равно придётся поддерживать старые версии
- ... или после пары итераций пользователи разбегутся

Управление изменениями API

В идеале:

- Сбор информации о текущих сценариях использования
- Анализ потенциального воздействия на пользователей
- Планирование изменений
- Обсуждение планов с пользователями
- Реализация

Управление изменениями API

Практика:

- Кто использует - ?
- Как использует - ?x?
- Оценка воздействия - ???



Управление изменениями API

Практика:

- Кто использует - ?
- Как использует - ?x?
- Оценка воздействия - ???



И что делать?

Управление изменениями API

Сбор общей информации

- Устраиваем опрос пользователей
- Анализируем историю баг-трекера

Управление изменениями API

Сбор общей информации

- Устраиваем опрос пользователей
- Анализируем историю баг-трекера

... всё очень приблизительно

Управление изменениями API

Сбор общей информации

- Устраиваем опрос пользователей
- Анализируем историю баг-трекера

... всё очень приблизительно

- Внутренний API - есть доступ к общей базе кода

Управление изменениями API

Сбор общей информации

- Устраиваем опрос пользователей
- Анализируем историю баг-трекера

... всё очень приблизительно

- Внутренний API - есть доступ к общей базе кода
... но как её анализировать?

Анализ API - инструменты

- IDEs, JArchitect, Structure101, Upsource
 - точно, но в пределах проекта
- jdeps, CDA, JDepend, ...
 - в пределах набора jar-файлов
 - зависимости с точностью до классов

Анализ API - инструменты

- IDEs, JArchitect, Structure101, Upsource
 - точно, но в пределах проекта
- jdeps, CDA, JDepend, ...
 - в пределах набора jar-файлов
 - зависимости с точностью до классов

**Существующим инструментам
не достаёт охвата!**

Анализ API – чего хотим?

- Искать использования API
 - Контекстно (как) в IDE
 - Во всех доступных проектах
- С учётом информации о версиях

Анализ API – чего хотим?

- Искать использования API
 - Контекстно (как) в IDE
 - Во всех доступных проектах
- С учётом информации о версиях

Спасение утопающих...

- LinkedIn - сделал JARVIS (для себя)
- JetBrains - тоже что-то сделал ...

Анализ API – чего хотим?

- Искать использования API
 - Контекстно (как) в IDE
 - Во всех доступных проектах
- С учётом информации о версиях

Спасение утопающих...

- LinkedIn - сделал JARVIS (для себя)
- JetBrains - тоже что-то сделал ...

... дело рук самих утопающих

dxLab presents...

USAGES

Usages

В двух словах:

- Поиск зависимостей над Maven хранилищем кода
- Не зависит от исходного языка*
- Учитывает версии анализируемых артефактов
- Интегрирован с IntelliJ IDEA

* Анализ не зависит, IDEA plugin пока поддерживает только Java

Настройка плагина

Configure Usages plugin

Usages server URLs +

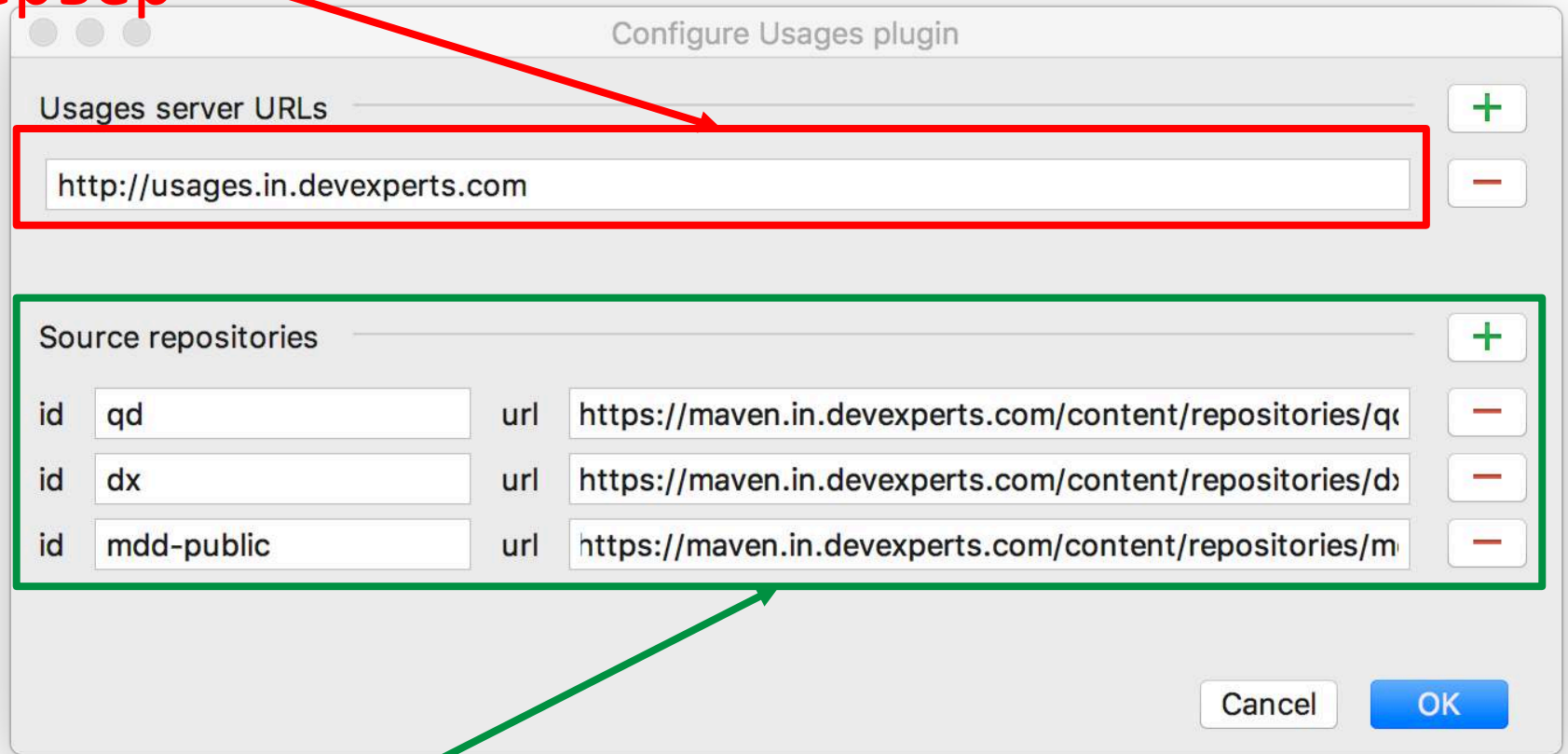
-

Source repositories +

id	<input type="text" value="qd"/>	url	<input type="text" value="https://maven.in.devexperts.com/content/repositories/qd"/>	-
id	<input type="text" value="dx"/>	url	<input type="text" value="https://maven.in.devexperts.com/content/repositories/dx"/>	-
id	<input type="text" value="mdd-public"/>	url	<input type="text" value="https://maven.in.devexperts.com/content/repositories/mdd-public"/>	-

Настройка плагина

Сервер



Где искать исходный код

Настраиваем поиск

Find Maven usages

Find Maven usages of `com.devexperts.util.TimePeriod#valueOf(String)`

Find class usages Find derived classes usages
 Find method usages Find derived methods usages
 Find field usages

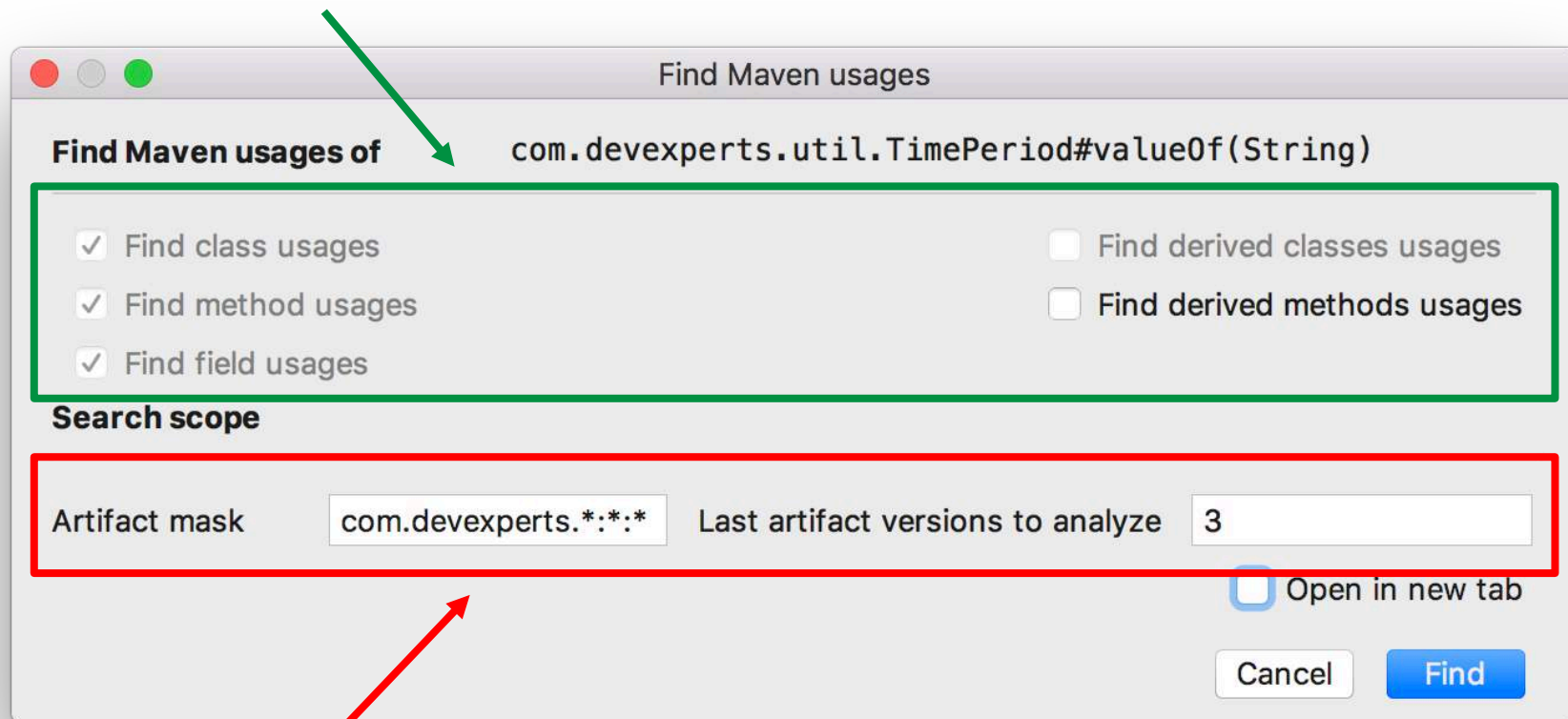
Search scope

Artifact mask Last artifact versions to analyze

Open in new tab

Настраиваем поиск

Уровень детализации



Где искать

Результат

Maven Usages of com.devexperts.util

- Found usages 6 usages
 - com.devexperts.util 6 usages
 - IndexedSet 2 usages
 - com.devexperts.qd:qd-core:3.256 1 usage
 - Field declaration 1 usage
 - com.devexperts.qd.core 1 usage
 - IndexedSet 1 usage
 - 35 com/devexperts/qd/core/MyIndexedSet 1 usage
- com.devexperts.qd:qd-core:3.257 1 usage
- TimePeriod 4 usages

6: TODO 9: Version Control Terminal Maven Usages 1 Event Log

Результат

Maven Usages of com.devexperts.util

- Found usages 6 usages
 - com.devexperts.util 6 usages
 - IndexedSet 2 usages
 - com.devexperts.qd:qd-core:3.256 1 usage
 - Field declaration 1 usage
 - com.devexperts.qd.core 1 usage
 - IndexedSet 1 usage
 - 35 com/devexperts/qd/core/MyIndexedSet 1 usage
- com.devexperts.qd:qd-core:3.257 1 usage
- TimePeriod 4 usages

6: TODO 9: Version Control Terminal Maven Usages 1 Event Log

Что ищем

Результат

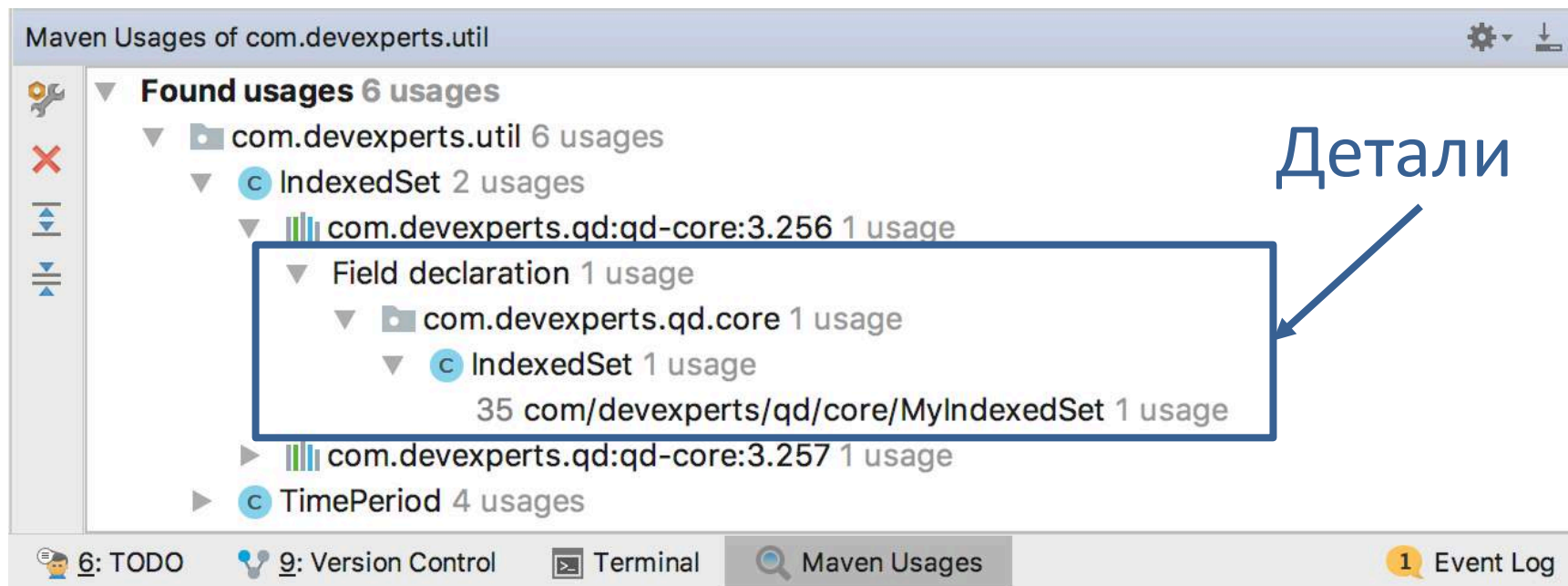
Maven Usages of com.devexperts.util

- Found usages 6 usages
 - com.devexperts.util 6 usages
 - IndexedSet 2 usages
 - com.devexperts.qd:qd-core:3.256 1 usage
 - Field declaration 1 usage
 - com.devexperts.qd.core 1 usage
 - IndexedSet 1 usage
 - 35 com/devexperts/qd/core/MyIndexedSet 1 usage
 - com.devexperts.qd:qd-core:3.257 1 usage
 - TimePeriod 4 usages

Где нашли

6: TODO 9: Version Control Terminal Maven Usages 1 Event Log

Результат



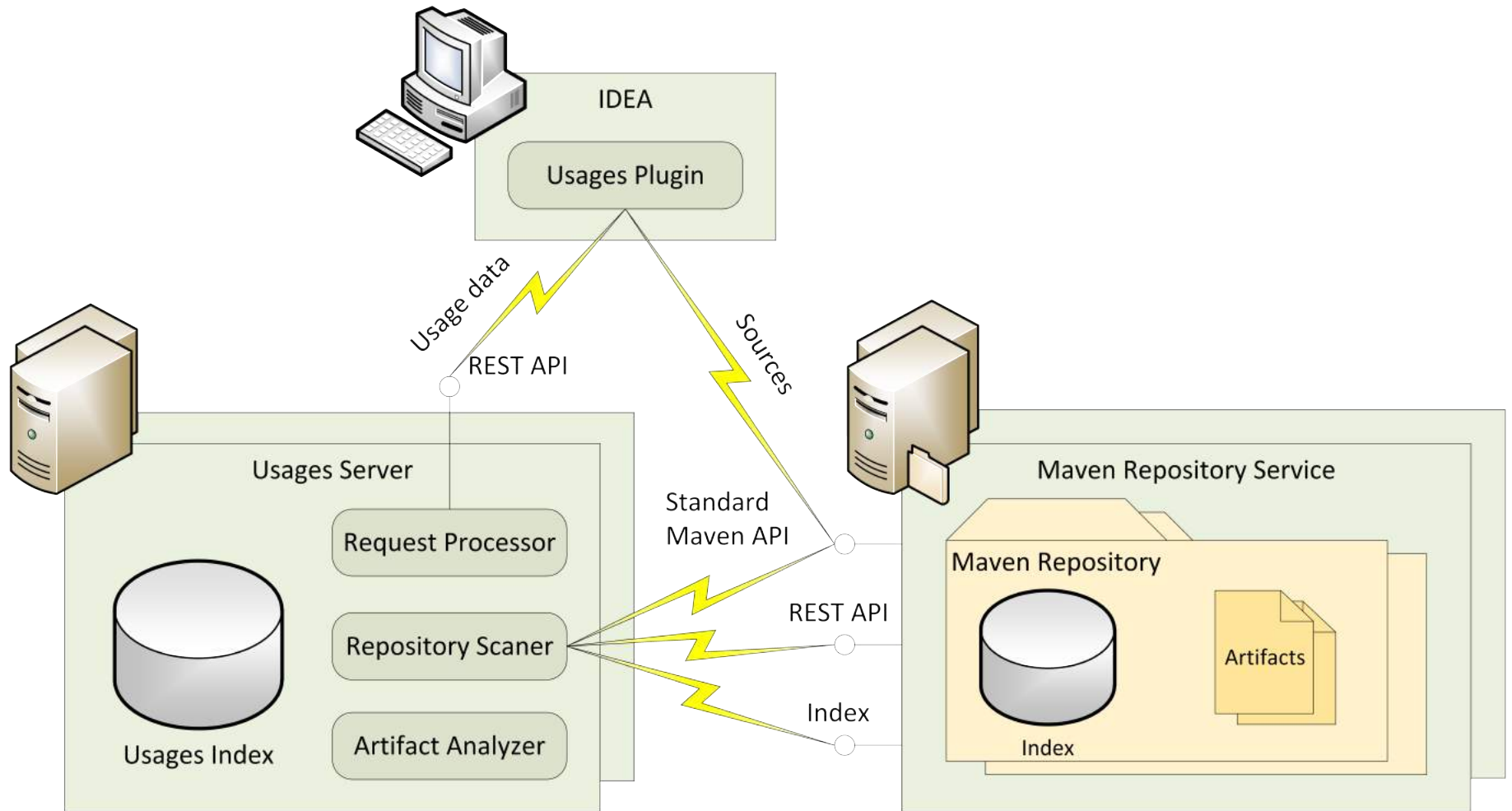
Maven Usages of com.devexperts.util

- Found usages 6 usages
 - com.devexperts.util 6 usages
 - IndexedSet 2 usages
 - com.devexperts.qd:qd-core:3.256 1 usage
 - Field declaration 1 usage
 - com.devexperts.qd.core 1 usage
 - IndexedSet 1 usage
 - 35 com/devexperts/qd/core/MyIndexedSet 1 usage

Детали

* Сопоставление с исходным кодом будет позже

Как всё устроено?



Сканирование Maven репозиториев

- Нельзя просто так взять, и просканировать Maven репозиторий
 - Нет универсального API
 - Nexus, Artifactory, Bintray - у каждого всё своё!*
- Nexus repository manager
 - REST API
 - Lucene индексы
 - Артефакты с maven-координатами
 - Список файлов в каждом артефакте

* Текущая версия поддерживает только Nexus

Алгоритм выполнения запроса

1. Определяем в каких артефактах может быть искомый элемент*
 - Знаем, какие пакеты содержатся в артефакте
2. Находим их зависимости
 - Анализируем pom-файлы при сканировании
3. Составляем список всех возможных использований
 - Используем ASM для анализа байт-кода
 - Все результаты анализа записываются в индексное хранилище
4. Отдаем клиенту запрошенные данные
 - С точностью до номера строки кода (если есть) или метода

* А еще смотрим на потомков классов, если нужно

Перспективы

- Статистические отчёты
- Сравнение версий API
- Поддержка модулей Java 9+
- Поддержка Artifactory, ...
- Анализ значений параметров (data-flow analysis)
- Обнаружение несовместимостей при сборке
- Анализ зависимостей с учётом истории изменений
 - Например, чтобы определить степень готовности проектов к переходу
 - ... или провести анализ изменения API со временем

Где взять?

<https://code.devexperts.com/display/USAGES>

<https://github.com/Devexperts/usages>

Где взять?

<https://code.devexperts.com/display/USAGES>

<https://github.com/Devexperts/usages>

Early Access Prototype

Спасибо за внимание!

Дмитрий Цителов

cit@devexperts.com

<https://www.linkedin.com/in/dmitrytsitelov>