

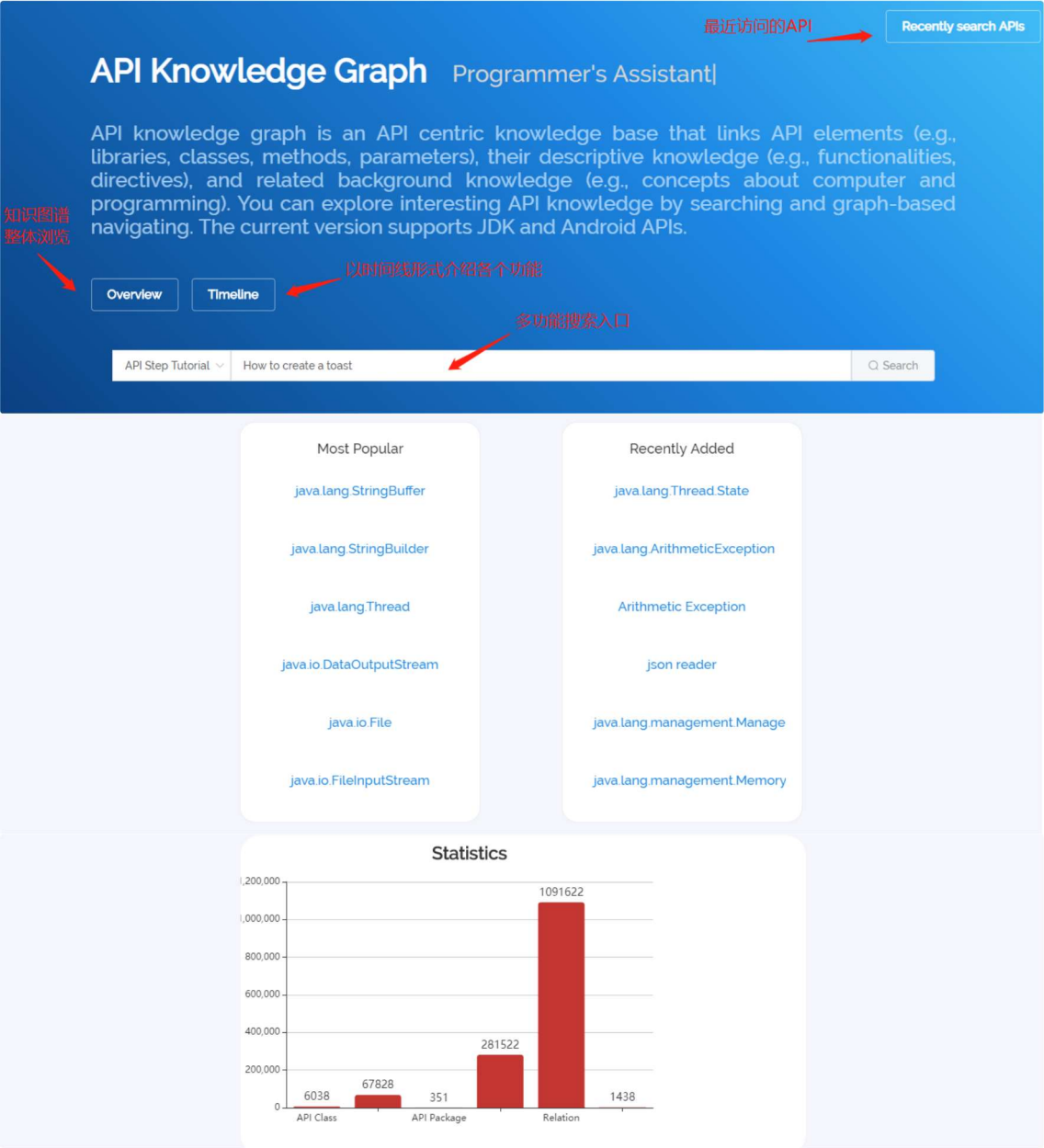
# API知识图谱平台用户指南

平台构建了API知识图谱，整合了CodeWisdom团队相关研究成果，为软件开发者和科研人员提供了API图浏览、多功能搜索、API知识汇总等服务，通过搜索和基于图的导航来探索有趣的API知识。当前版本支持JDK和Android API。

## 首页功能概览

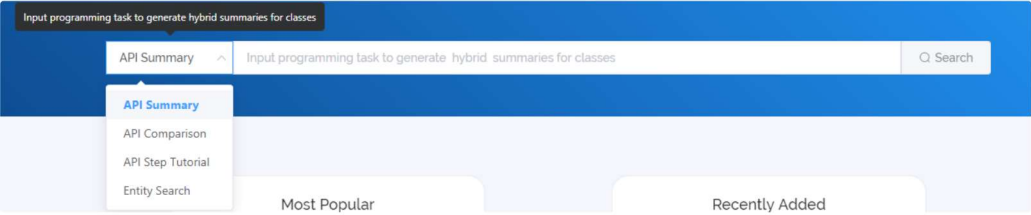
首页功能概览如下图所示：

- 1. 上方显示了知识图谱的整体浏览入口、平台的发展时间线，多功能搜索、最近访问的API历史记录。
- 2. 中间是最常访问和最新添加的实体、数据规模统计。
- 3. 最下方是单独的系统工具，包括了根据编程任务生成API摘要、相似API对比、根据编程任务生成API使用教程。





# 多功能搜索



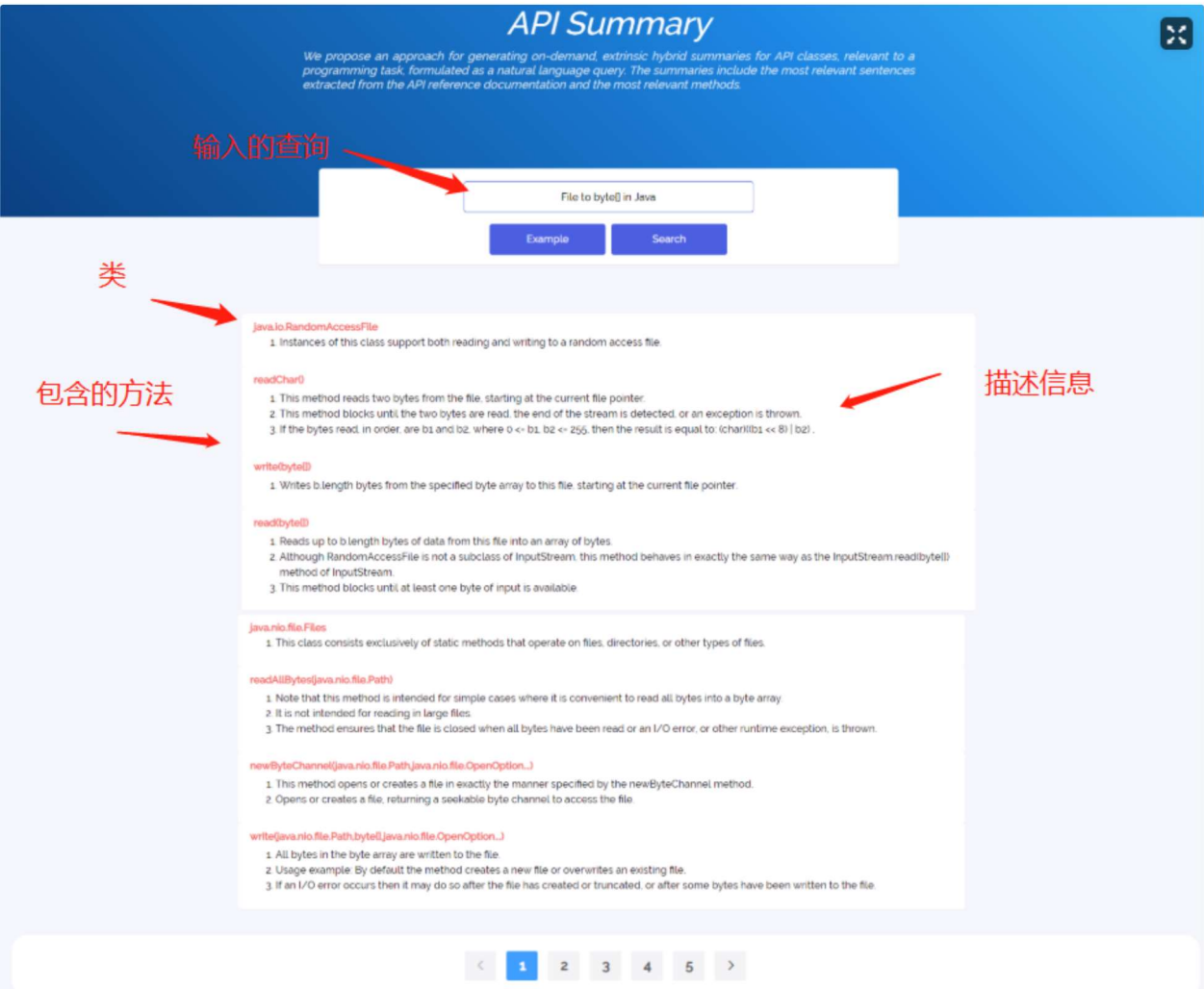
多功能搜索目前支持4种类型的搜索：

- API Summary: 输入一个编程任务，搜索类级别的摘要信息。
- API Comparison: 输入2个API，生成在功能、特性、概念上的对比结果。
- API Step Tutorial: 输入一个编程任务，生成解决该任务的API教程。
- Entity Search: 在知识图谱中搜索实体信息。

点击search后，会根据问题类型，返回不同的结果。

## API摘要 ( API Summary )

输入一个编程任务，生成的摘要是以类为核心，从API参考文档中提取的最相关的句子以及最相关的方法。



## API比较 ( API Comparison )

会根据用户输入的查询内容，匹配出所比较的APIs，从功能（functionality）、特性（characteristic）、概念（Concept Classification）、从属（Membership）这些角度，产生比较结果。

# API Comparison

We propose a knowledge graph-based approach APIComp that automatically extracts API comparison knowledge from API reference documentation to support the comparison of a pair of API classes or methods from different aspects.

**Concept Classification:** Describe that an API element is an instance of a category by concept.  
**Membership:** Describe that an API element belongs to a category.  
**characteristic:** Describe the characteristics of an API element, its property or compare the characteristics of two API elements, their properties or functionalities.  
**functionality:** Describe what an API element can or can not do, or compare the functionalities of two API elements by three relations. (equivalent to, similar to, different from)

java.lang.StringBuffer

java.lang.StringBuilder

Search

Example1

Example2

java.lang.StringBuffer	java.lang.StringBuilder
character sequence	
buffers	builder
string buffer capacity	string builder capacity
string buffer initial capacity	string builder instances
serializable	
modifiable	

concept

classification

membership

characteristic

functionality

## API教程（API Step Tutorial）

根据用户查询场景，返回一系列的相关教程来解决编程困难，以动作为核心，辅以样例代码。

# API Step Tutorial

Accomplishing a program task usually involves performing multiple activities in a logical order. Task-solving activities may have different relationships, such as subactivityof, precede-follow, and different attributes, such as location, condition, API, code. We refer to task-solving activities and their relationships and attributes as know-how knowledge. Programming task know-how knowledge is commonly documented in semi-structured textual tutorials. Published paper on related work.

How to create a toast

Example

Search

## Creating a Custom Toast View

- create a customized layout for your toast notification .
- define a View layout
- pass the root View object to the setContentView method .
- retrieve the LayoutInflater
- inflate the layout
- use this inflated layout to find more View objects
- create a new Toast
- set some properties of the toast
- call setContentView and pass it the inflated layout .
- display the toast

```
LayoutInflater inflater = getLayoutInflater();
View layout = inflater.inflate(R.layout.custom_toast,
    (ViewGroup) findViewById(R.id.custom_toast_container));

TextView text = (TextView) layout.findViewById(R.id.text);
text.setText("This is a custom toast");

Toast toast = new Toast(getApplicationContext());
toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);
toast.setDuration(Toast.LENGTH_LONG);
toast.setView(layout);
toast.show();
```

步骤

示例代码

## The Basics

- **instantiate** a Toast object
- **display** the toast notification

\$\$23982\$\$

```
Context context = getApplicationContext();
CharSequence text = &quot;Hello toast!&quot;;
int duration = Toast.LENGTH_SHORT;
```

```
Toast toast = Toast.makeText(context, text, duration);
toast.show();
```

## Adding behavior to individual items

- use **setOnClickListener** LRB RRB to **set** an object's click behavior such as to cause a button to launch an Activity .
- use **setOnClickListener** LRB RRB .
- **setting** a fill
- **is** the index LRB position RRB of the touched view .

```
public class StackWidgetService extends RemoteViewsService {
    <span style="color: crimson;">@Override</span>
    public RemoteViewsFactory onGetViewFactory(Intent intent) {
        return new StackRemoteViewsFactory(this.getApplicationContext(), intent);
    }
}

class StackRemoteViewsFactory implements RemoteViewsService.RemoteViewsFactory {
    private static final int mCount = 10;
    private List<WidgetItem> mWidgetItems = new ArrayList<WidgetItem>();
    private Context mContext;
    private int mAppWidgetId;

    public StackRemoteViewsFactory(Context context, Intent intent) {
        mContext = context;
        mAppWidgetId = intent.getIntExtra(AppWidgetManager.EXTRA_APPWIDGET_ID,
            AppWidgetManager.INVALID_APPWIDGET_ID);
    }

    <span style="color: red;">/** Initialize the data set.</span>
    public void onCreate() {
        <span style="color: red;">/** In onCreate() you set up any connections / cursors to your data source.
        <span style="color: red;">/** for example downloading or creating content etc, should be deferred to
        <span style="color: red;">/** or getViewAt(). Taking more than 20 seconds in this call will result in
        for (int i = 0; i < mCount; i++) {
            mWidgetItems.add(new WidgetItem(i + &quot;!&quot;));
        }
        ...
    }
    ...

    <span style="color: red;">/** Given the position (index) of a WidgetItem in the array, use the item's text
    <span style="color: red;">/** combination with the app widget item XML file to construct a RemoteViews ob
    public RemoteViews getViewAt(int position) {
        <span style="color: red;">/** position will always range from 0 to getCount() - 1.</span>

        <span style="color: red;">/** Construct a RemoteViews item based on the app widget item XML file, and
        <span style="color: red;">/** text based on the position.</span>
        RemoteViews rv = new RemoteViews(mContext.getPackageName(), R.layout.widget_item);
        rv.setTextViewText(R.id.widget_item, mWidgetItems.get(position).text);

        <span style="color: red;">/** Next, set a fill-intent, which will be used to fill in the pending inter
        <span style="color: red;">/** that is set on the collection view in StackWidgetProvider.</span>
        Bundle extras = new Bundle();
        extras.putInt(StackWidgetProvider.EXTRA_ITEM, position);
        Intent fillInIntent = new Intent();
        fillInIntent.putExtras(extras);
        <span style="color: red;">/** Make it possible to distinguish the individual on-click</span>
        <span style="color: red;">/** action of a given item</span>
        rv.setOnClickListener(R.id.widget_item, fillInIntent);

        ...

        <span style="color: red;">/** Return the RemoteViews object.</span>
        return rv;
    }
    ...
}
```

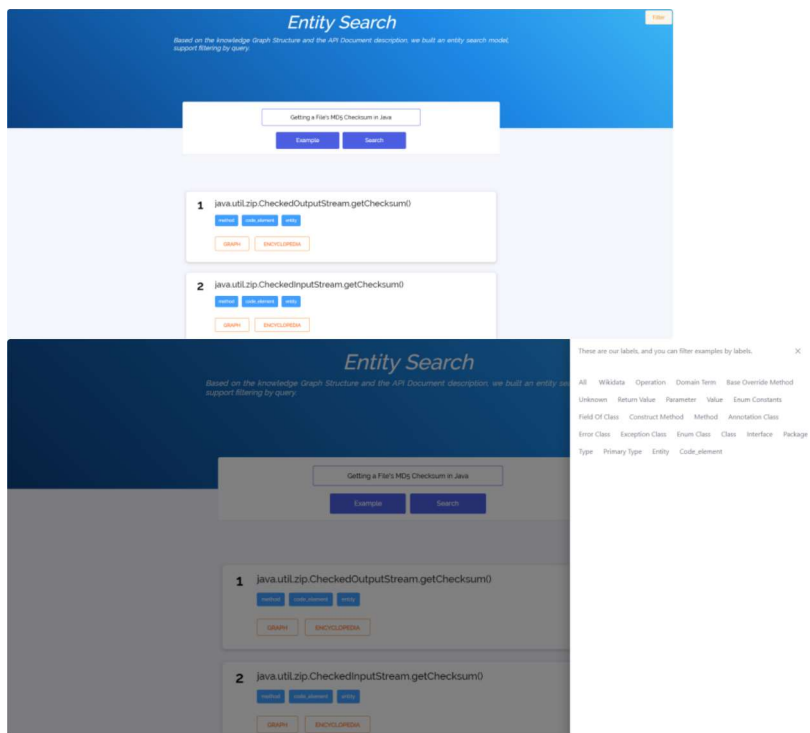
## Toasts overview

- **disappear after** a timeout .
- **clicking** Send
- **consider instead using** a Notification .

## 实体搜索 ( Entity Search )

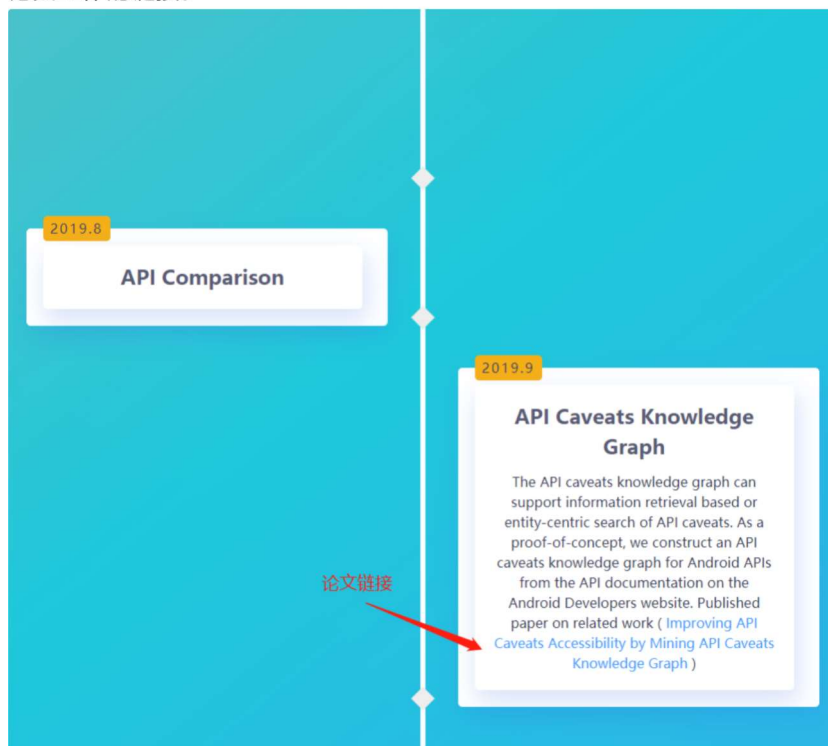
实体搜索会根据用户输入，在知识图谱中匹配最相关的节点，右上角点击**filter**可以对内容进行过滤。





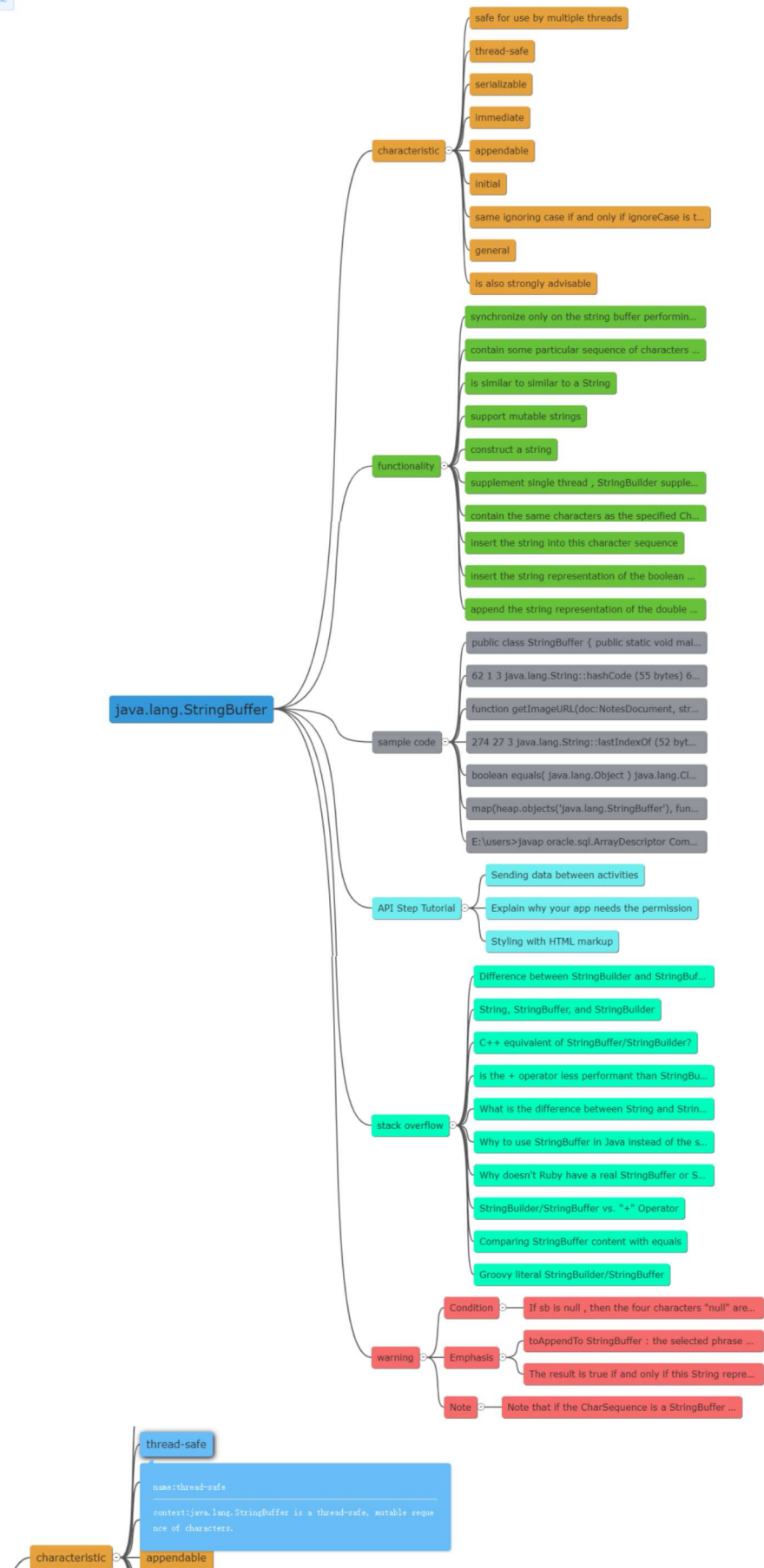
## 平台时间线

平台时间线自上而下介绍了平台功能的每一个里程碑，点击标题，会跳转到对应的功能模块，在描述介绍中，也有相关论文工作的链接。

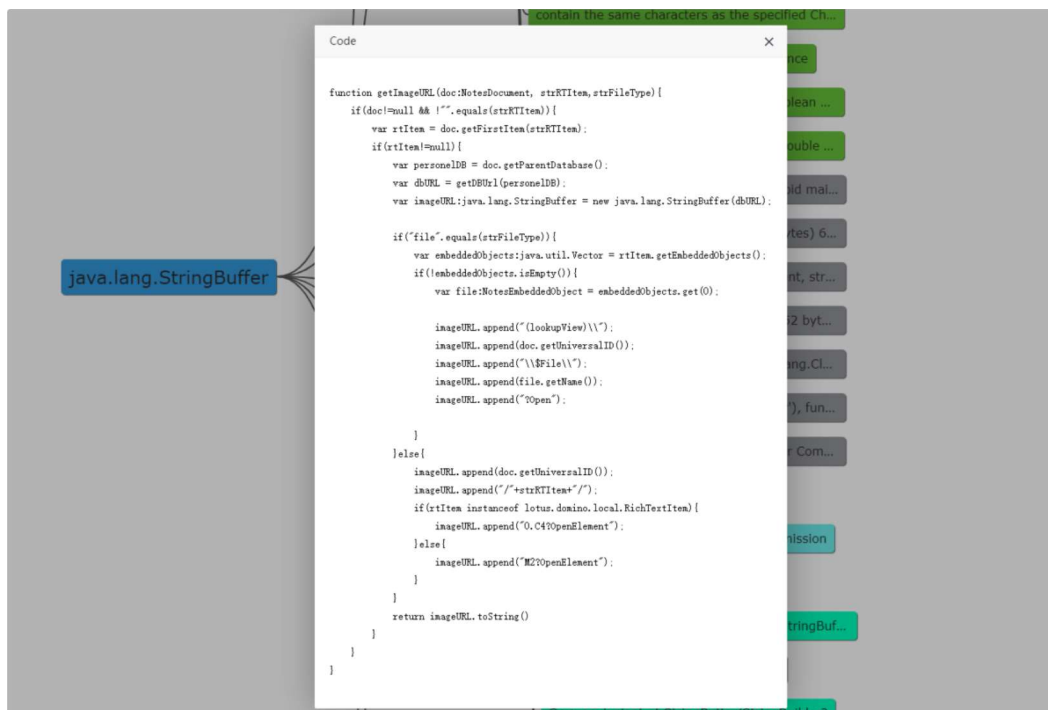


## API思维导图

为了对API多个层面的知识更好的汇总，采用API思维导图的形式展示。分为了API特性，功能、样例代码、使用教程、相关Stack Overflow讨论、使用警告或约束。思维导图上的内容，点击或者鼠标悬停，就显示出详细信息或者来源等。



悬停在特征上，显示来源



点击样例代码，显示完整内容

## 百科界面

百科界面内容分为：关系、图谱概览、词云、使用警告或约束信息、描述信息。

### 关系：

关系以[start, relation, end]三元组的形式显示，默认显示缩略结果，点击more会显示全部关系。

Table of Content:  
Relations  
Knowledge Graph  
Associated concept  
Warning  
Properties  
alias  
full\_declaration  
qualified\_name  
short\_description

**java.time**  
package

Relations

Start	Relation	End
java.time	belong to	java
java.time	mention in short description	main api
java.time	mention in short description	api
java.time	name related to	time
java.time.format	belong to	java.time

more

Knowledge Graph

点击此处，进入思维导图

### 图谱概览：

以选中的节点为中心，可以看到与它关联的其他API，如果更深层次的探索，可以点击"Graph Detail"。





点击词云中的一个概念，可以查看与这个概念相关的其他API。

点击词云中的一个概念，可以查看与这个概念相关的其他API。

```
java.time.format.DateTimeFormatterBuilder.append(ZoneOffset.UTC)

android.icu.util.Calendar.computeZoneOffset

java.time.OffsetTime.MIN

java.xml.datatype.XMLGregorianCalendar.getTimeZone(int)

java.util.Calendar.ZONE_OFFSET

java.time.OffsetDateTime.MIN

java.xml.datatype.DateTimeConstants.MAX_TIMEZONE_OFFSET

java.time.format.DateTimeFormatterBuilder.appendInstant()

android.icu.util.Calendar.computeZoneOffset(appendInstant)

java.time.format.DateTimeFormatterBuilder.appendInstant()
```

[illegible]

tag表示信息的类型，如强调（Emphasis），提示（Note），条件（Condition）等。

tag	text
Emphasis	toAppendTo StringBuffer : the selected phrase will be appended to this StringBuffer.
Emphasis	The result is true if and only if this String represents the same sequence of characters as the specified StringBuffer.
Note	Note that if the CharSequence is a StringBuffer then the method synchronizes on it.

tag	text
Condition	If sb is null , then the four characters "null" are appended to this StringBuffer.

描述信息显示名称信息和段描述。

alias :

## String Buffer

## StringBuffer

java.lang.StringBuffer

qualified\_name :

java.lang.StringBuffer

short\_description :

A thread-safe, mutable sequence of characters.