

Requirement

TrackPoint 项目方案说明

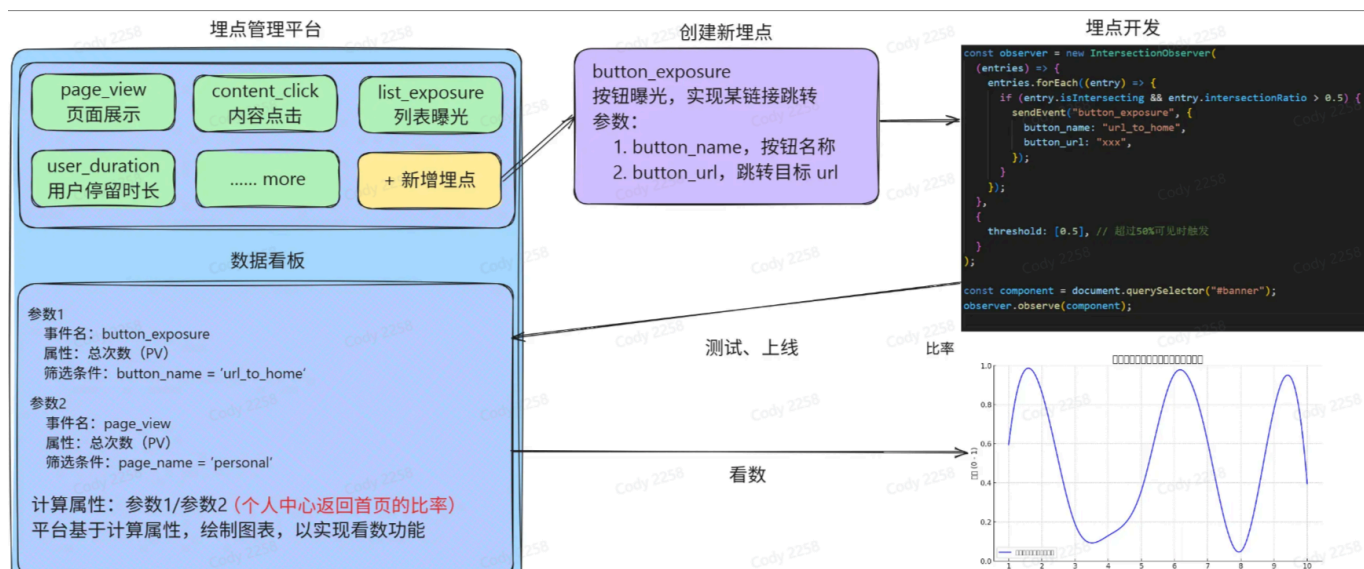
一、项目背景

什么是埋点？

埋点是一种用于跟踪用户在网站或应用中行为的数据采集技术，通过记录点击、浏览等用户行为，帮助团队进行用户行为分析、项目AB实验、C端行为监听，同时有助于指导优化方向和资源分配。本项目旨在开发一个完整的埋点研发体系，提供项目用户行为分析、性能监控、报警监控的能力。

二、项目要求

从一个完整的埋点研发流程来看，涉及了三个方向：埋点 SDK、埋点数据服务、埋点平台，涉及到的方面较多，因此仅要求完成最核心的功能，相关内容如下：



• 埋点 SDK :

• 功能要求 :

- 埋点事件的上报：实现一个基本的埋点事件上报，如：

```
import { sendEvent } from '@byte/track-point'
sendEvent(EventName.XXX_EVENT, {
  params1: xxx1,
  params2: xxx2
})
```

- 页面初始化：在页面初始化时，注册相关的埋点信息，如项目 id、采样频率等，如：

```
import { register } from '@byte/track-point'
register({
  project_id: 'xxx',
  upload_percent: 0.1,
  // .....
})
```

- 添加通用参数：可以在任意时刻，添加埋点上报时默认带上的参数，如：

```
import { addCommonParams } from '@byte/track-point'
addCommonParams({
  common1: 'common1',
  common2: 'common2',
})
```

- 获取用户基本信息：初始化时，自动读取用户的环境信息，如浏览器版本、操作系统、uid、上报时间等一系列数据
- 错误信息捕获：当接入项目出现报错时，自动捕获相关信息并进行上报
- 技术要求：
 - 技术栈：推荐使用 **ts + node** 实现
- 加分项：可以尝试拓展更多的功能，如支持：同时上报请求的最大限制、埋点合并上报队列的等待时间、白屏监控等功能
- 埋点数据服务：(**server**)
 - 功能要求：
 - 设计较为完善的数据库表结构，并尝试通过用于支撑服务端端侧的数据存取
 - 设计合理的索引加速数据的查询
 - 基于事件埋点的多样性，合理运用相关的设计模式，打造出易拓展，低耦合的埋点事件 crud
 - 技术要求：
 - 技术栈：推荐使用 **node** 实现 **server** 服务（其余实现也可以，这里不做任何限定）
 - 加分项：要求对数据存取的性能做优化策略（主要解决数据查询慢，半结构化数据存储）
- 埋点平台：(数据看板、埋点管理)
 - 功能要求：
 - 埋点数据的查询：可以获取指定埋点事件的 PV/UV（PV：次数，UV：人数）数据等
 - 事件信息的增删改查：可以对埋点事件进行管理，涉及基本的增删改查能力
 - 埋点数据的筛选：可以实现对 **xx** 事件的条件筛选，如：**xx** 事件且 **params1 = 'xxx'** 的 PV
 - 看板搭建：可以搭建可视化的数据图表，支持折线图、饼图等各种样式，同时支持日期、事件，用户行为的筛选
 - 技术要求：
 - 技术栈：推荐使用 **React** 或 **Vue** 实现
 - 加分项：提供优秀的看板搭建机制、精确的筛选能力等

三、详细设计

目标	说明
基础功能 (必须)	<p>埋点 SDK</p> <ol style="list-style-type: none">埋点事件的上报：实现基本的埋点事件上报功能，能将指定的事件名称及对应参数发送到服务端。页面初始化：在页面加载初始化阶段，注册项目相关的埋点基础信息，为后续数据上报提供配置依据。添加通用参数：允许在任意时刻为埋点上报添加默认携带的通用参数，增强数据的丰富性和关联性。获取用户环境信息：在初始化时自动读取用户的环境信息，如浏览器版本、操作系统、uid、上报时间等，随埋点数据一同上报，帮助更全面地分析用户行为。错误信息捕获：当接入项目出现报错时，自动捕获相关错误信息并上报给服务端，便于及时发现和排查问题。 <p>埋点数据服务 (server)</p> <ol style="list-style-type: none">数据库表结构设计：设计合理完善的数据库表结构，满足对埋点数据的高效存储以及方便后续的查询、分析等操作需求索引设计：通过合理创建索引，加速对埋点数据的查询操作，提高系统整体性能。埋点事件 CRUD 设计：基于事件埋点的多样性，运用合适的设计模式打造易于拓展、低耦合的埋点事件增删改查功能，方便对数据进行管理和维护 <p>埋点平台 (数据看板、埋点管理)</p> <ol style="list-style-type: none">埋点数据的查询：能够获取指定埋点事件的 PV (访问次数)、UV (独立访客人数) 等关键数据指标，为分析用户行为提供基础数据支持。事件信息的增删改查：提供对埋点事件的基本管理功能，包括新增、删除、修改、查询事件相关信息，方便对埋点规则和配置进行维护。埋点数据的筛选：实现对埋点事件按照特定条件进行筛选查询，比如查询满足某个参数条件的事件的 PV 等，满足更精细化的数据分析需求。看板搭建：搭建可视化的数据图表，支持折线图、饼图等多种样式，同时能基于日期、事件、用户行为等维度进行筛选展示，直观呈现数据分析结果。
进阶功能 (可选)	<p>埋点 SDK</p> <ol style="list-style-type: none">埋点合并上报队列的等待时间：通过设置一定时间间隔，将收集到的埋点事件先缓存起来，再按间隔批量合并发送，减少上报请求次数。同时上报请求的最大限制：为避免过多并发上报请求导致性能问题或服务端压力过大，设置可配置的同时上报请求数量上限。白屏监控：实时监测页面是否出现白屏情况，一旦检测到长时间空白或未正常渲染状态，及时上报相关信息，助力快速定位页面加载问题。 <p>埋点数据服务 (server)</p> <ol style="list-style-type: none">数据查询优化：针对数据查询速度慢的问题，综合运用多种策略优化查询操作，提高数据获取效率，保障系统响应性能。半结构化数据存储优化：妥善处理埋点事件参数这类半结构化数据的存储问题，在保证数据存储合理、高效的同时，方便后续查询操作，提升整体数据管理效能。

目标	说明
	<p>埋点平台（数据看板、埋点管理）</p> <p>1. 优秀的看板搭建机制：提供更为灵活、便捷且功能强大的看板搭建功能，支持用户根据多样化需求自主定制可视化数据展示效果，提升数据分析的针对性与实用性。</p> <p>2. 精确的筛选能力：增强数据筛选功能的精准度与灵活性，支持复杂的多维度筛选条件组合。</p>
探索性功能 (不作实现要求)	<p>埋点 SDK</p> <p>1. 高可用性：确保 SDK 在面临诸如网络不稳定、应用异常等各类状况时，依然能够保障埋点数据的完整性，并维持尽可能持续稳定的上报能力，提升整体服务的可靠性。</p> <p>2. 高健壮性：让 SDK 具备强大的应对异常情况的能力，使其在不同浏览器版本、操作系统环境以及可能出现的资源限制等复杂条件下，均能稳定运行。</p> <p>埋点数据服务（server）</p> <p>3. 高可用性：确保埋点数据服务在遭遇各类故障场景时，能够迅速恢复正常运行状态，最大限度降低对数据采集、分析流程的影响，保障系统可以持续稳定地为业务提供支持。</p> <p>4. 高健壮性：使埋点数据服务具备强大的容错能力，能够妥善应对诸如网络波动、硬件故障、软件异常等多种复杂异常情况。</p> <p>埋点平台（数据看板、埋点管理）</p> <p>5. 高可用性：保障埋点平台在面对各种可能导致服务中断或异常的情况时，能够快速恢复正常，维持数据查询、展示以及管理等功能的持续可用，为用户提供可靠的数据分析服务。</p> <p>6. 高健壮性：使埋点平台能够妥善处理各种异常输入、错误操作以及系统内部突发问题，保证在复杂的实际使用场景下稳定、准确地完成数据展示、分析与管理等功能。</p>

四、评分标准

基本评分标准

评价项	评分说明	分数
功能实现	- 基础功能是否完整 - 业务项目能否顺利接入TrackPoint 并实现基本功能	60
代码质量	- 项目结构清晰，代码符合编码规范 - 注释完整，变量命名规范	15
sdk 测试	- 是否包含单元测试 - 测试覆盖率是否达到一定标准，关键功能是否经过充分验证	15

评价项	评分说明	分数
性能	- 性能符合基本要求，不出现明显的卡顿现象 - 在一些场景做出性能优化策略	10

加分项 (最多 20 分)

- 完成进阶功能、探索性功能
- 举例功能外，提出更多自己的想法并实现
- 优秀的平台交互设计

五、更多信息

DEMO 介绍

为了让大家能更清晰的了解埋点 SDK 的概念，这里提供了基本事件上报的最小实现 DEMO，仅供参考：

```
const eventName = {
  CLICK_EVENT: 'click_event',
  PAGE_VIEW_EVENT: 'page_view_event'
};

// @byte/tracker-point 模拟命名空间
const { sendEvent } = {
  sendEvent: function (eventName, params) {
    // 构建要发送到服务器的数据对象，包含事件名称和参数等信息
    const dataToSend = {
      event_name: eventName,
      event_params: params
    };

    const apiUrl = 'https://xxx-xxx-api-url.com/tracker-point';

    return fetch(apiUrl, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(dataToSend)
    })
      .then(response => response.json())
      .then(result => {
        console.log('埋点事件上报结果:', result);
        return result;
      })
      .catch(error => {
```

```

        console.error('埋点事件上报出错:', error);
        throw error;
    });
}
};

sendEvent(EventName.CLICK_EVENT, {
    button_id: 'button_1',
    page_url: window.location.href
});

```

平台参考

- 阿里云可观测 (<https://sls.aliyun.com/doc/>)
- 阿里云CloudLens For SLS (<https://help.aliyun.com/zh/sls/user-guide/cloudlens-for-sls/>)



