

k kb-honor-admin

办选率统计分析

主要实现英雄禁用率和选择率数据统计分析。

首页 玩家列表 办选数据

排名	英雄	登场率	禁用率	Actions
1	百里玄策	15%	5%	查看详情
2	孙悟空	30%	15%	查看详情
3	鲁班七号	42%	1%	查看详情
4	后羿	43%	2%	查看详情
5	王昭君	35%	12%	查看详情
6	貂蝉	12%	5%	查看详情
7	钟馗	32%	10%	查看详情
8	牛魔	18%	6%	查看详情
9	亚瑟	15%	3%	查看详情
10	吕布	29%	7%	查看详情

共 10 条 20条/页 1 前往 1 页



英雄办选信息查询

列表显示英雄办选数据

数据类型声明, api/types.d.ts

```
export interface BanPick {  
  hero: Hero,  
  ban: number,  
  pick: number  
}
```

办选数据, mock/heros.ts

```
import { BanPick } from "../src/api/types";  
const banpick: BanPick[] = [  
  { hero: heros[0], ban: 0.05, pick: 0.15 },  
  { hero: heros[1], ban: 0.15, pick: 0.3 },  
  { hero: heros[2], ban: 0.01, pick: 0.42 },  
  { hero: heros[3], ban: 0.02, pick: 0.43 },  
  { hero: heros[4], ban: 0.12, pick: 0.35 },  
  { hero: heros[5], ban: 0.05, pick: 0.12 },  
  { hero: heros[6], ban: 0.1, pick: 0.32 },  
  { hero: heros[7], ban: 0.06, pick: 0.18 },  
  { hero: heros[8], ban: 0.03, pick: 0.15 },  
  { hero: heros[9], ban: 0.07, pick: 0.29 },  
]
```

接口声明

```
/banpick:  
  get:  
    description: 查询英雄办选统计数据列表  
    operationId: getBanPickData  
    responses:  
      '200':  
        description: OK
```

接口实现, mock/heros.ts

```

export const getBanPickData = (req: Request, res: Response) => {
  const { page = 1, limit = 20 } = req.query
  const pageList = banpick.filter(
    (_, index) => index < limit * page && index >= limit * (page - 1))

  return res.json({
    code: 20000,
    data: {
      banpick: pageList,
      total: banpick.length
    }
  })
}

```

新增办选信息路由, @/router/modules/heros.ts

```

import { RouteConfig } from 'vue-router'
import Layout from '@layout/index.vue'

export const heroRoutes: RouteConfig = {
  path: '/heros',
  component: Layout,
  redirect: '/heros/list',
  meta: {
    title: 'heroMgt',
    icon: 'qq'
  },
  children: [
    {
      path: 'banpick',
      component: () => import('@views/hero/banpick.vue'),
      name: 'BanPick',
      meta: {
        title: 'banPick',
        icon: 'like'
      }
    }
  ]
}

```

新增办选列表页面, 创建@/views/hero/banpick.vue

```
<template>
```

```

<div class="app-container">
  <el-table
    v-loading="listLoading"
    :data="list"
    border
    fit
    highlight-current-row
    style="width: 100%"
  >
    <el-table-column
      align="center"
      label="排名"
    >
      <template v-slot="{ $index }">
        <span>{{ $index + 1 }}</span>
      </template>
    </el-table-column>

    <el-table-column
      align="center"
      label="英雄"
    >
      <template v-slot="{ row }">
        <span>{{ row.hero.name }}</span>
      </template>
    </el-table-column>

    <el-table-column
      align="center"
      label="登场率"
    >
      <template v-slot="{ row }">
        <span>{{ row.pick }}</span>
      </template>
    </el-table-column>

    <el-table-column
      align="center"
      label="禁用率"
    >
      <template v-slot="{ row }">
        <span>{{ row.ban }}</span>
      </template>
    </el-table-column>
  </el-table>

  <pagination
    v-show="total>0"
    :total="total"
  >

```

```

        :page.sync="listQuery.page"
        :limit.sync="listQuery.limit"
        @pagination="getList"
      />
    </div>
  </template>

  <script lang="ts">
    import { Component, Vue } from 'vue-property-decorator'
    import { getBanpick } from '@/api/heros'
    import { BanPick, BanPickDetail } from '@/api/types'
    import Pagination from '@/components/Pagination/index.vue'

    @Component({
      name: 'BanPick',
      components: {
        Pagination
      }
    })
    export default class extends Vue {
      private total = 0;
      private list: BanPick[] = [];
      private listLoading = true;
      private listQuery = {
        page: 1,
        limit: 20,
      };

      created() {
        this.getList()
      }

      private async getList() {
        this.listLoading = true
        const { data } = await getBanpick(this.listQuery)
        this.list = data.banpick
        this.total = data.total
        this.listLoading = false
      }

      private handleFilter() {
        this.listQuery.page = 1
        this.getList()
      }
    }
  </script>

```

办选详情

获取某个英雄详细办选信息，并使用线图展示。

模拟数据，mock/heros.ts

```
const banpickDetail = [
  {
    hero: heros[0],
    week: {
      ban: [0.05, 0.04, 0.03, 0.05, 0.06, 0.05, 0.05],
      pick: [0.15, 0.1, 0.11, 0.12, 0.09, 0.12, 0.11]
    },
    month: {
      ban: [0.05, 0.04, 0.03, 0.05, 0.06, 0.05, 0.05, 0.05, 0.04, 0.03,
0.05, 0.06, 0.05, 0.05, 0.05, 0.04, 0.03, 0.05, 0.06, 0.05, 0.05, 0.05, 0.04,
0.03, 0.05, 0.06, 0.05, 0.05],
      pick: [0.15, 0.1, 0.11, 0.12, 0.09, 0.12, 0.11, 0.05, 0.04, 0.03,
0.05, 0.06, 0.05, 0.05, 0.05, 0.04, 0.03, 0.05, 0.06, 0.05, 0.05, 0.05, 0.04,
0.03, 0.05, 0.06, 0.05, 0.05]
    },
    quarter: {
      ban: [0.05, 0.04, 0.03, 0.05],
      pick: [0.15, 0.1, 0.11, 0.12]
    },
    year: {
      ban: [0.05, 0.04, 0.03, 0.05, 0.06, 0.05, 0.05, 0.03, 0.05, 0.06,
0.05, 0.05],
      pick: [0.15, 0.1, 0.11, 0.12, 0.09, 0.12, 0.11, 0.1, 0.11, 0.12,
0.09, 0.10]
    }
  }
]
```

数据类型，api/types.d.ts

```

export interface Bp {
  ban: number[],
  pick: number[]
}
export type RangeType = 'week' | 'month' | 'quarter' | 'year'
export type BanPickDetail = {
  [key in RangeType]: Bp
}
export type BanPickDetailAndHero = { hero: Hero } & BanPickDetail

```

接口声明, swagger.yml

```

/banpick/{id}:
  get:
    description: 查询指定英雄的办选详情数据
    operationId: getBanPickDetail
    responses:
      '200':
        description: OK

```

接口实现, mock/heros/ts

```

export const getBanPickDetail = (req: Request, res: Response) => {
  const { id } = req.params
  const detail = banpickDetail.find(item => item.hero.id === +id)
  console.log(detail);

  return res.json({
    code: 20000,
    data: {
      detail
    }
  })
}

```

新增"办选详情"路由, router/modules/heros.ts

```

{
  path: 'banpick/:id(\\d+)',
  component: () => import('@views/hero/banpick-detail.vue'),
  name: 'BanPickDetail',
  meta: {
    title: 'banPickDetail',
    icon: 'chart',
    activeMenu: '/heros/banpick',
    hidden: true
  }
}

```

导航至详情页，banpick.vue

```

<template>
  <el-table-column
    align="center"
    label="Actions"
    width="120"
  >
    <template v-slot="{row}">
      <router-link :to="'/heros/banpick/'+row.hero.id">
        <el-button type="primary">查看详情</el-button>
      </router-link>
    </template>
  </el-table-column>
</template>

```

获取办选详情数据

请求，@/api/heros

```

export const getBanpickDetail = (id: string) =>
  request(`/banpick/${id}`)

```

获取数据，banpick-detail.vue

```

import { Component, Vue } from 'vue-property-decorator'
import { getBanpickDetail } from '@/api/heros'
import { BanPickDetailAndHero } from '@/api/types'

@Component({
  name: 'BanPickDetail'
})
export default class extends Vue {

```



```

    detail: BanPickDetailAndHero | null = null;

    async created() {
      const res = await getBanpickDetail(this.$route.params.id)
      this.detail = res.data.detail
    }
  }
}

```

数据展示, @/views/hero/components/LineChart.vue

```

<template>
  <div :class="className" :style="{height: height, width: width}" />
</template>

<script lang="ts">
import echarts, { EChartOption } from "echarts";
import { Component, Prop, Watch } from "vue-property-decorator";
import { mixins } from "vue-class-component";
import ResizeMixin from "@/components/Charts/mixins/resize";

// 修改数据接口
export interface ILineChartData {
  heroName: string,
  xAxisData: string[],
  banData: number[],
  pickData: number[];
}

@Component({
  name: "LineChart"
})
export default class extends mixins(ResizeMixin) {
  @Prop({ required: true }) private chartData!: ILineChartData;
  @Prop({ default: "chart" }) private className!: string;
  @Prop({ default: "100%" }) private width!: string;
  @Prop({ default: "350px" }) private height!: string;

  @Watch("chartData", { deep: true })
  private onChartDataChange(value: ILineChartData) {
    this.setOptions(value);
  }

  mounted() {
    this.$nextTick(() => {
      this.initChart();
    });
  }

  beforeDestroy() {

```

```

    if (!this.chart) {
        return;
    }
    this.chart.dispose();
    this.chart = null;
}

private initChart() {
    this.chart = echarts.init(this.$el as HTMLDivElement, "macarons");
    this.setOptions(this.chartData);
}

private setOptions(chartData: ILineChartData) {
    if (this.chart) {
        this.chart.setOption({
            backgroundColor: "#394056",
            title: {
                top: 20,
                text: this.chartData.heroName + "近期办选走势",
                textStyle: {
                    fontWeight: "normal",
                    fontSize: 16,
                    color: "#F1F1F3"
                },
                left: "1%"
            },
            tooltip: {
                trigger: "axis",
                formatter: ([f1, f2]: EChartOption.Tooltip.Format[]) => {
                    const v1 = f1.data * 100 + "%";
                    const v2 = f2.data * 100 + "%";
                    return (
                        "办选率<br>" +
                        f1.marker +
                        f1.seriesName +
                        ":" +
                        v1 +
                        "<br>" +
                        f2.marker +
                        f2.seriesName +
                        ":" +
                        v2
                    );
                }
            },
            legend: {
                top: 20,
                icon: "rect",
                itemWidth: 14,

```

```

    itemHeight: 5,
    itemGap: 13,
    data: ["禁用率", "登场率"],
    right: "4%",
    textStyle: {
      fontSize: 12,
      color: "#F1F1F3"
    }
  },
  grid: {
    top: 100,
    left: "2%",
    right: "2%",
    bottom: "2%",
    containLabel: true
  },
  xAxis: [
    {
      type: "category",
      boundaryGap: false,
      axisLine: {
        lineStyle: {
          color: "#57617B"
        }
      },
      data: this.chartData.xAxisData
    }
  ],
  yAxis: [
    {
      type: "value",
      name: "百分比",
      axisTick: {
        show: false
      },
      axisLine: {
        lineStyle: {
          color: "#57617B"
        }
      },
      axisLabel: {
        margin: 10,
        fontSize: 14,
        formatter: function(value: number) {
          return value * 100 + "%";
        }
      },
      splitLine: {
        lineStyle: {

```

```

        color: "#57617B"
      }
    }
  },
  series: [
    {
      name: "登场率",
      type: "line",
      // smooth: true,
      symbol: "circle",
      symbolSize: 5,
      // showSymbol: false,
      lineStyle: {
        width: 2
      },
      areaStyle: {
        color: new echarts.graphic.LinearGradient(
          0,
          0,
          0,
          1,
          [
            {
              offset: 0,
              color: "rgba(0, 136, 212, 0.3)"
            },
            {
              offset: 0.8,
              color: "rgba(0, 136, 212, 0)"
            }
          ]
        ),
        shadowColor: "rgba(0, 0, 0, 0.1)",
        shadowBlur: 10
      },
      itemStyle: {
        color: "rgb(0,136,212)",
        borderColor: "rgba(0,136,212,0.2)",
        borderWidth: 12
      },
      data: this.chartData.pickData
    },
    {
      name: "禁用率",
      type: "line",
      // smooth: true,
      symbol: "circle",

```

```

        symbolSize: 5,
        // showSymbol: false,
        lineStyle: {
            width: 2
        },
        areaStyle: {
            color: new echarts.graphic.LinearGradient(
                0,
                0,
                0,
                1,
                [
                    {
                        offset: 0,
                        color: "rgba(219, 50, 51, 0.3)"
                    },
                    {
                        offset: 0.8,
                        color: "rgba(219, 50, 51, 0)"
                    }
                ],
                false
            ) as any,
            shadowColor: "rgba(0, 0, 0, 0.1)",
            shadowBlur: 10
        },
        itemStyle: {
            color: "rgb(219,50,51)",
            borderColor: "rgba(219,50,51,0.2)",
            borderWidth: 12
        },
        data: this.chartData.banData
    }
]
} as EChartOption<EChartOption.SeriesLine>;
}
}
}
</script>

```

显示图表, banpick-detail.vue

```

<template>
  <div class="chart-container">
    <line-chart v-if="detail" height="100%" :chart-data="this.chartData" />
    <p v-else>没有任何数据...</p>
  </div>

```

```

</template>
<script lang="ts">
//引入图表
import LineChart from "../components/LineChart.vue";
//RangeType: 时间区间类型
//RangeData: 时间区间数据
import { RangeType, RangeData } from "@api/types";
import { getDateArray } from "@utils/gen-month-date";

//RangeData: 时间区间数据
const rangeData: RangeData = {
  week: ["周一", "周二", "周三", "周四", "周五", "周六", "周日"],
  month: new Array(31).fill(1).map((v, i) => (i+1) + ''),
  quarter: ["一季度", "二季度", "三季度", "四季度"],
  year: new Array(12).fill(1).map((v, i) => (i+1) + '月')
};

@Component({
  // 添加图表
  components: {
    LineChart
  }
})
export default class extends Vue {
  // 默认时间区间是周
  range: RangeType = "week";

  // 计算属性: 根据detail和range获取显示数据内容
  get chartData() {
    // 数据不存在时
    if (!this.detail) {
      return { heroName: "", xAxisData: [], banData: [], pickData: [] };
    }
    // 数据存在或发生变化时
    const xAxisData = rangeData[this.range];
    const banData = this.detail[this.range].ban;
    const pickData = this.detail[this.range].pick;
    const data = { heroName: this.detail.hero.name, xAxisData, banData,
    pickData }

    return data;
  }
}
</script>

```

时间区间发生变化

```

<template>
  <div class="chart-container">
    <!-- 周期选择 -->
    <div class="range">
      <el-radio-group v-model="range">
        <el-radio-button label="week">周</el-radio-button>
        <el-radio-button label="month">月</el-radio-button>
        <el-radio-button label="quarter">季度</el-radio-button>
        <el-radio-button label="year">年</el-radio-button>
      </el-radio-group>
    </div>
    ...
  </div>
</template>

<style lang="scss" scoped>
.range {
  padding: 20px;
  position: absolute;
  z-index: 10;
  left: -138px;
  width: 275px;
  margin: 0 50%;
}
.chart-container {
  position: relative;
  width: 100%;
  height: calc(100vh - 84px);
}
</style>

```