EdgeModel 文档

`EdgeModel` 用于描述工作流编辑器或绘图应用中的连线数据结构。
它定义了连接的起止节点、样式、动画配置、标签等信息，采用不可变数据结构，便于扩展、维护及序列化。

一、EdgeModel 基础信息

字段名	类型	说明
`id`	`String`	连线的唯一标识
`sourceNodeId`	`String`	源节点ID
`sourceAnchorId`	`String`	源锚点ID
`targetNodeId`	`String?`	目标节点ID（可为空）
`targetAnchorId`	`String?`	目标锚点ID（可为空）
`isConnected`	`bool`	连线是否已连接（默认：false）
`edgeType`	`String`	连线类型（例如："flow", "dependency"）

二、状态与协作信息（可选）

字段名	类型	说明
`status`	`String?`	连线状态（例如："running", "error"）
`locked`	`bool`	是否被锁定（禁止编辑或移动）
`lockedByUser`	`String?`	锁定此连线的用户ID（多人协作时使用）
`version`	`int`	连线版本号
`zIndex`	`int`	连线渲染的层级（越高越靠上）

三、样式属性 (`EdgeLineStyle`)

定义连线外观的纯数据：

字段名	类型	说明
`colorHex`	`String?`	连线颜色（"#RRGGBB" 或 "#AARRGGBB"）
`strokeWidth`	`double`	线条宽度（默认：3.0）
`dashPattern`	`List<double>?`	虚线模式（例如：[5, 2]）
`lineCap`	`EdgeLineCap`	线条端点样式（butt, round, square）

字段名	类型	说明
lineJoin	EdgeLineJoin	拐角样式（miter, round, bevel）
arrowStart	ArrowType	起点箭头类型
arrowEnd	ArrowType	终点箭头类型
arrowSize	double	箭头大小（默认：10.0）
arrowAngleDeg	double	箭头夹角（默认：30.0 度）

四、动画配置 (EdgeAnimationConfig)

字段名	类型	说明
animate	bool	是否启用动画
animationType	String?	动画类型（例如："flowPulse", "dashBlink"）
animationSpeed	double?	动画速度
animationPhase	double?	动画相位
animationColorHex	String?	动画颜色
animateDash	bool	是否启用虚线流动效果
dashFlowPhase	double?	虚线流动的相位
dashFlowSpeed	double?	虚线流动速度

五、路径与标签（可选）

字段名	类型	说明
waypoints	List<List<double>>?	连线的路径点坐标（例如：[[100, 100], [200, 200]]）
label	String?	连线上的文字标签
labelStyle	Map<String, dynamic>?	标签的样式信息

六、额外数据

字段名	类型	说明
data	Map<String, dynamic>	存储自定义业务数据

七、核心方法说明

(1) copyWith

不可变更新，返回新实例：

```
final newEdge = edge.copyWith(
  status: 'completed',
  targetNodeId: 'node2',
);
```

(2) toJson & fromJson

序列化与反序列化：

```
final json = edge.toJson();
final edge = EdgeModel.fromJson(json);
```

(3) EdgeLineStyle 与 EdgeAnimationConfig

EdgeLineStyle 与 EdgeAnimationConfig 是 EdgeModel 的子类，用于描述连线的样式与动画配置。

```
final lineStyle = EdgeLineStyle(
  colorHex: '#FF0000',
  strokeWidth: 2.0,
  dashPattern: [5, 2],
  arrowEnd: ArrowType.triangle,
);

final animConfig = EdgeAnimationConfig(
  animate: true,
  animationType: 'flowPulse',
  animationSpeed: 1.0,
);
```

(4) EdgeModel 的构造方法

```
final edge = EdgeModel(
  id: 'edge1',
  sourceNodeId: 'node1',
  sourceAnchorId: 'anchor1',
  targetNodeId: 'node2',
  targetAnchorId: 'anchor2',
  isConnected: true,
  edgeType: 'flow',
```

```
  status: 'running',
);
```

八、使用场景示例

1. 连线状态更新

```
final updatedEdge = edge.copyWith(
  status: 'completed',
  targetNodeId: 'node2',
);
```

2. 连线样式更新

```
final updatedEdge = edge.copyWith(
  lineStyle: EdgeLineStyle(colorHex: '#00FF00'),
);
```

3. 连线动画配置

```
final updatedEdge = edge.copyWith(
  animConfig: EdgeAnimationConfig(animate: false),
);
```

九、枚举类型定义

```
enum EdgeLineCap { butt, round, square }

enum EdgeLineJoin { miter, round, bevel }

enum ArrowType { none, triangle, diamond, arrow }
```

十、JSON 示例

```
{
  "id": "edge1",
  "sourceNodeId": "node1",
  "sourceAnchorId": "anchor1",
  "targetNodeId": "node2",
  "isConnected": true,
```

```
  "edgeType": "flow",
  "status": "running",
  "locked": false,
  "version": 1,
  "zIndex": 1,
  "waypoints": [[100, 100], [200, 200]],
  "lineStyle": {
    "colorHex": "#FF0000",
    "strokeWidth": 2.0,
    "arrowEnd": "triangle"
  },
  "animConfig": {
    "animate": true,
    "animationType": "flowPulse",
    "animationSpeed": 1.0
  },
  "label": "Flow Edge",
  "labelStyle": {
    "fontSize": 14
  },
  "data": {
    "customKey": "value"
  }
}
```

十一、单元测试建议

- 测试构造方法默认值与边界值
- 测试copyWith方法的不可变性
- 测试JSON序列化与反序列化是否一致
- 测试getAnchorOffset返回准确的坐标值

十二、总结

EdgeModel 以不可变数据结构实现，便于扩展、维护及序列化。通过合理设计状态与数据的边界，能够很好地适用于复杂工作流或绘图应用。