提醒频率功能方案说明文档

一、功能背景

为 "添加提醒" 页面扩展提醒频率选项（支持每天 / 每周 / 每月 / 每季度 / 每年 / 一次），并实现首页自动筛选今日待执行的提醒。

二、核心数据结构设计

在 reminders 本地缓存集合中，为每个提醒增加 frequency（频率类型） 和 frequencyConfig（频率配置） 字段，用于存储提醒规则：

字段名	类型	说明
id	String	提醒唯一 ID（如 r_123456）
familyId	String	关联的家人 ID
content	String	提醒内容（如 "每周一散步"）
remindTime	String	提醒时间（如 "08:30"）
frequency	String	频率类型（枚举值：daily/ weekly/monthly/quarterly/ yearly/once）
frequencyConfig	Object	频率对应的配置项（不同频率结构不同）
createDate	String	创建日期（格式：YYYY-MM-DD）

各频率对应的 frequencyConfig 示例

frequency 类型	frequencyConfig 配置示例	说明
daily（每天）	{}	无额外配置
weekly（每周）	{ "weekdays": [1, 3] }	每周一、周三（1 = 周一）
monthly（每月）	{ "days": [5, 20] }	每月 5 日、20 日
quarterly（每季度）		1/4/7/10 月的 15 日

frequency 类型	frequencyConfig 配置示例	说明
	{ "months": [1,4,7,10], "days": [15] }	
yearly（每年）	{ "month": 10, "day": 1 }	每年 10 月 1 日
once（一次）	{ "date": "2024-01-01" }	仅 2024 年 1 月 1 日执行一次

三、"添加提醒" 页面实现

1. 页面交互逻辑

通过 条件渲染，不同频率自动显示对应的配置项（如选择 "每周" 则显示 "选择星期" 的选择器）。

2. WXML 代码（add-reminder.wxml）

```
<view class="reminder-frequency">
  <picker bindchange="onFrequencyChange" value="{{frequencyIndex}}"
range="{{frequencyRange}}">
    <view class="picker">
  提醒频率： {{frequencyRange[frequencyIndex]}}
    </view>
  </picker>

  <!-- 每周配置 -->
  <view wx:if="{{frequency === 'weekly'}}">
    <picker mode="multiSelector" bindchange="onWeekdayChange"
value="{{selectedWeekdaysIndex}}" range="{{weekdayRange}}">
      <view class="picker">
   选择星期： {{selectedWeekdaysNames.length > 0 ?
selectedWeekdaysNames.join('、') : '请选择'}}
      </view>
    </picker>
  </view>

  <!-- 每月配置 -->
  <view wx:if="{{frequency === 'monthly'}}">
    <picker mode="multiSelector" bindchange="onDayChange" value="{{selectedDaysIndex}}"
range="{{dayRange}}">
      <view class="picker">
   选择日期： {{selectedDays.length > 0 ? selectedDays.join('、') : '请选
择'}}
```

```xml
        </view>
      </picker>
    </view>

    <!-- 每年配置 -->
    <view wx:if="{{frequency === 'yearly'}}">
      <picker mode="date" bindchange="onYearlyDateChange" value="{{selectedDate}}"
  start="{{today}}">
        <view class="picker">
   选择日期: {{selectedDate || '请选择'}}
        </view>
      </picker>
    </view>

    <!-- 一次配置 -->
    <view wx:if="{{frequency === 'once'}}">
      <picker mode="date" bindchange="onOnceDateChange" value="{{selectedDate}}"
  start="{{today}}">
        <view class="picker">
   选择日期: {{selectedDate || '请选择'}}
        </view>
      </picker>
    </view>
  </view>
```

3. JS 代码（add-reminder.js）

```
const dayjs = require('../../utils/dayjs');

Page({
  data: {
    // 频率选择列表
    frequencyRange: ['每天', '每周', '每月', '每年', '一次'],
    frequencyIndex: 0,
    frequency: 'daily',

    // 每周配置
    weekdayRange: [['周日', '周一', '周二', '周三', '周四', '周五', '周六']],
    selectedWeekdaysIndex: [],
    selectedWeekdaysNames: [],

    // 每月配置
    dayRange: [[...Array(31).keys()].map(i => (i + 1).toString())],
    selectedDaysIndex: [],
```

```
    selectedDays: [],

    // 日期配置
    today: dayjs().format('YYYY-MM-DD'),
    selectedDate: '',

    // 其他表单字段
    content: '',
    remindTime: '08:00',
    familyId: ''
  },

  // 切换频率类型
  onFrequencyChange(e) {
    const index = e.detail.value;
    const frequencyMap = ['daily', 'weekly', 'monthly', 'yearly', 'once'];

    this.setData({
      frequencyIndex: index,
      frequency: frequencyMap[index],
      // 重置其他频率的配置
      selectedWeekdaysIndex: [],
      selectedDaysIndex: [],
      selectedDate: ''
    });
  },

  // 选择每周的星期
  onWeekdayChange(e) {
    const value = e.detail.value;
    const selectedNames = value.map(i => this.data.weekdayRange[0][i]);
    const selectedWeekdays = value.map(i => (i === 0 ? 7 : i)); // 周日转7

    this.setData({
      selectedWeekdaysIndex: value,
      selectedWeekdaysNames: selectedNames,
      frequencyConfig: { weekdays: selectedWeekdays }
    });
  },

  // 选择每月的日期
  onDayChange(e) {
    const value = e.detail.value;
    const selectedDays = value.map(i => parseInt(this.data.dayRange[0][i]));

    this.setData({
      selectedDaysIndex: value,
```

```
        selectedDays: selectedDays,
        frequencyConfig: { days: selectedDays }
      });
    },

    // 选择每年的日期
    onYearlyDateChange(e) {
      const dateStr = e.detail.value;

      this.setData({
        selectedDate: dateStr,
        frequencyConfig: {
          month: dayjs(dateStr).month() + 1,
          day: dayjs(dateStr).date()
        }
      });
    },

    // 选择一次的日期
    onOnceDateChange(e) {
      this.setData({
        selectedDate: e.detail.value,
        frequencyConfig: { date: e.detail.value }
      });
    },

    // 保存提醒
    saveReminder() {
      const { frequency, frequencyConfig, content, remindTime, familyId }
  = this.data;

      // 表单校验
      if (!content) {
        wx.showToast({
          title: '请输入提醒内容',
          icon: 'none'
        });
        return;
      }

      // 生成新提醒
      const newReminder = {
        id: `r_${Date.now()}`,
        familyId,
        content,
        remindTime,
        frequency,
```

```javascript
      frequencyConfig: frequencyConfig || {},
      createDate: dayjs().format('YYYY-MM-DD')
    };

    // 存入缓存
    let reminders = wx.getStorageSync('reminders') || [];
    reminders.push(newReminder);
    wx.setStorageSync('reminders', reminders);

    wx.showToast({
      title: '添加成功'
    });
    wx.navigateBack();
  }
});
```

四、首页筛选逻辑

1. 筛选工具函数（utils/reminder-filter.js）

```
const dayjs = require('./dayjs');

// 判断单条提醒是否为今日待执行
function isReminderForToday(reminder) {
  const today = dayjs();
  const { frequency, frequencyConfig } = reminder;

  switch (frequency) {
    case 'daily':
      return true;
    case 'weekly':
      const todayWeekday = today.day() === 0 ? 7 : today.day();
      return frequencyConfig.weekdays?.includes(todayWeekday);
    case 'monthly':
      return frequencyConfig.days?.includes(today.date());
    case 'quarterly':
      const isQuarterMonth = [1,4,7,10].includes(today.month() + 1);
      return isQuarterMonth && frequencyConfig.days?.includes(today.date());
    case 'yearly':
      return frequencyConfig.month === (today.month() + 1) &&
frequencyConfig.day === today.date();
    case 'once':
      return today.isSame(frequencyConfig.date, 'day');
```

```javascript
      default:
        return false;
    }
  }

  // 筛选今日待执行的所有提醒
  function filterTodayReminders(allReminders) {
    return (allReminders || []).filter(reminder => isReminderForToday(reminder));
  }

  module.exports = { filterTodayReminders };
```

2. 首页使用（index.js）

```javascript
const { filterTodayReminders } = require('../../utils/reminder-filter');

Page({
  data: {
    todayReminders: []
  },

  onLoad() {
    this.loadTodayReminders();
  },

  loadTodayReminders() {
    const allReminders = wx.getStorageSync('reminders') || [];
    const todayReminders = filterTodayReminders(allReminders);
    this.setData({ todayReminders });
  }
});
```

五、功能优势

1. **配置灵活**：不同频率的规则通过 frequency 和 frequencyConfig 解耦，支持后续扩展更多频率；
2. **体验友好**：添加提醒时自动显示对应配置项，减少用户操作成本；
3. **性能高效**：首页仅需遍历一次提醒，通过工具函数快速筛选，无性能压力；
4. **代码可维护**：核心逻辑抽离为工具函数，页面代码更简洁。