**private** Object view; 用于描述视图信息

**private** ModelMap model; 用于描述模型数据(响应数据)

**public** **void** setViewName(String viewName) 设置视图名字

**public** ModelAndView addObject(String attributeName, Object attributeValue) 设置模型数据

**protected** Map<String, Object> getModelInternal() 返回模型数据

**public** ModelMap getModelMap() 返回模型数据

**public** Map<String, Object> getModel() 返回模型数据

2. ModelAndView 处理模型数据源码调试:

[1]. 在 DispatcherServlet中的 945 行, 开始调用请求处理器中的请求处理方法



[2]. 执行请求处理器中的请求处理方法 ,方法执行完成后，返回一个ModelAndView对象, 最终将ModelAndView对象返回到

DispatcherServlet的945行.

[3]. 在DispatcherServlet的959行， 准备开始处理ModelAndView.

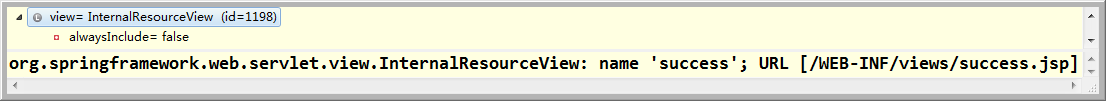


[4]. 在DispatcherServlet 的 1012行，准备处理ModelAndView中的视图信息还有模型数据.



[5]. 在DispatcherServlet的1204行，根据ModelAndView中的视图信息，通过视图解析器 解析得到一个View视图对象.





[6]. 在DispatcherServlet的1225行， 视图对象开始处理模型数据



解释: render()方法是View接口中定义的。 在 AbstractView类中进行了实现. 所有具体的视图类都使用AbstractView中实现的render方法.

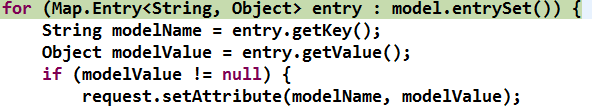
[7]. 在AbstractView中的266行，整合输出模型数据.



解释: renderMergedOutputModel方法是 AbstractView中定义的抽象方法， 在每个具体的视图实现类中进行实现.

[8]. 在InternalResourceView中的180行, 将模型数据设置到Request域对象中.





[9].在InternalResourceView中的189行， 获取转发器



[10].在InternalResourceView中的209行， 开始转发

