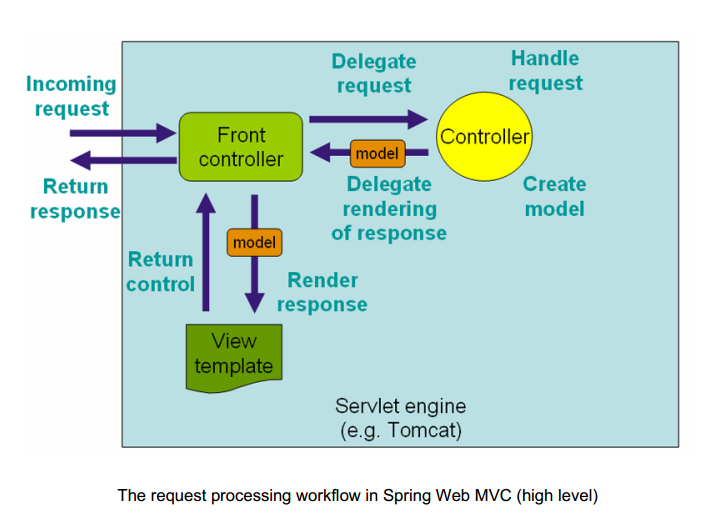
SpringMVC流程：



DispatcherServlet

request

handlerMapping

Handler(controller)

Detect

handler

handlerExecutionChain

Handleradapter

doDispatch

modelandview

initStrategies(ApplicationContext context){

initMultipartResolver(context);

initLocaleResolver(context);

initThemeResolver(context);

initHandlerMappings(context);

initHandlerAdapters(context);

initHandlerExceptionResolvers(context);

initRequestToViewNameTranslator(context);

initViewResolvers(context);

initFlashMapManager(context);

}

doDispatch{

// Determine handler for the current request.

mappedHandler = ~~getHandler~~(processedRequest, **false**) {

HandlerMapping hm

HandlerExecutionChain handler = hm.getHandler(request);

}

HandlerAdapter ha = getHandlerAdapter(mappedHandler.getHandler());

// Actually invoke the handler.

mv = ha.handle(processedRequest, response, mappedHandler.getHandler());

processDispatchResult(processedRequest, response, mappedHandler, mv, dispatchException){

render(mv, request, response){

view = resolveViewName(mv.getViewName(), mv.getModelInternal(), locale, request);

view.render(mv.getModelInternal(), request, response){

abstractview.render(Map<String, ?> model, HttpServletRequest request, HttpServletResponse response){

renderMergedOutputModel(mergedModel, request, response){

exposeModelAsRequestAttributes(model, requestToExpose){

**if** (modelValue != **null**) {

request.setAttribute(modelName, modelValue);

}

}

}

}

}

}

}