

-----list out a page data only in a quiz --- == -----

```
from rest_framework.pagination import PageNumberPagination

class CustomPagination(PageNumberPagination):
    page_size = 10 # 10 items per page
    page_size_query_param = 'page_size'
    max_page_size = 100

class GetQuizzesByPageRangeView(APIView):
    pagination_class = CustomPagination

    def get(self, request, start_page, end_page):
        try:
            # Convert to integers and validate
            start_page = int(start_page)
            end_page = int(end_page)

            if start_page < 1 or end_page < start_page:
                return Response(
                    {"error": "Invalid page range. Start page
must be less than or equal to end page"},
                    status=status.HTTP_400_BAD_REQUEST
                )

            # Get base queryset
            queryset = Quiz.objects.all().order_by('-created_at')

            # Initialize paginator
            paginator = self.pagination_class()

            results = []
            total_quizzes = queryset.count()
            page_size = paginator.get_page_size(request)

            # Calculate total pages
            total_pages = (total_quizzes + page_size - 1) //
page_size

            # Adjust end_page if it exceeds total pages
            end_page = min(end_page, total_pages)

            # Get quizzes for the specified page range
            for page_num in range(start_page, end_page + 1):
                # Get the page
                page = paginator.paginate_queryset(queryset,
request, page=page_num)
                if not page:
                    break # No more pages available

            # Serialize the page
```

```

        serializer = QuizSerializer(page, many=True)

        # Add page number to each quiz
        for quiz in serializer.data:
            quiz['page_number'] = page_num
            results.append(quiz)

    if not results:
        return Response([],
            status=status.HTTP_200_OK
        )

    return Response({
        'status': 'success',
        'count': total_quizzes,
        'page_size': page_size,
        'total_pages': total_pages,
        'start_page': start_page,
        'end_page': end_page,
        'results': results
    })

except ValueError:
    return Response(
        {"error": "Page numbers must be integers"},
        status=status.HTTP_400_BAD_REQUEST
    )

except Exception as e:
    return Response(
        {"error": f"An error occurred: {str(e)}"},
        status=status.HTTP_500_INTERNAL_SERVER_ERROR
    )

```